Scripts to populate marc_indexers version

MODSOURCE-601 introduces changes to improve write operations into the "marc_records_lb" table. These changes require executing scripts to populate the newly introduced "marc_indexers"."version" column.

The scripts should be executed separately from each other and sequentially (each one in a separate transaction).

1. Delete marc_indexers related to OLD records (no need to run if it was executed at least once)

   ```sql
   DELETE FROM {tenantId}_mod_source_record_storage.marc_indexers
   WHERE exists(
       SELECT 1
       FROM {tenantId}_mod_source_record_storage.records_lb
       WHERE records_lb.id = marc_indexers.marc_id
       AND records_lb.state = 'OLD'
   );
   ```

2. Set marc_indexers version

   ```sql
   UPDATE {tenantId}_mod_source_record_storage.marc_indexers
   SET version = 0
   WHERE version IS NULL;
   ```

3. Populate marc_records_tracking

   ```sql
   INSERT INTO {tenantId}_mod_source_record_storage.marc_records_tracking
   SELECT id, 0, false
   FROM {tenantId}_mod_source_record_storage.marc_records_lb
   LEFT JOIN {tenantId}_mod_source_record_storage.marc_records_tracking ON marc_records_tracking.marc_id = marc_records_lb.id
   WHERE marc_records_tracking.marc_id IS NULL;
   ```

4. Recreate marc_indexers indexes

   No need to run when upgrading straight to mod-source-record-storage v5.6.11 or greater.
Recreate marc_indexers indexes

do $$
declare
    index integer;
    suffix text;
begin
    for index in 0 .. 999 loop
        suffix = lpad(index::text, 3, '0');
        execute 'drop index if exists {tenantId}_mod_source_record_storage.idx_marc_indexers_marc_id_' || suffix || ';
        raise info 'Starting to create index "{tenantId}_mod_source_record_storage.idx_marc_indexers_marc_id_version_{%}"', suffix;
        execute 'create index if not exists idx_marc_indexers_marc_id_version_{%} on {tenantId}_mod_source_record_storage.marc_indexers_{%}(marc_id, version);'
        commit;
        raise info 'Index "{tenantId}_mod_source_record_storage.idx_marc_indexers_marc_id_version_{%}" has been created', suffix;
    end loop;
end;
$$;

5. Create constraint

Create constraint

do $$
declare
    index integer;
    suffix text;
begin
    for index in 0 .. 999 loop
        suffix = lpad(index::text, 3, '0');
        raise info 'Starting to create not-null constraint on "{tenantId}_mod_source_record_storage.idx_marc_indexers_%_version" column', suffix;
        execute 'ALTER TABLE {tenantId}_mod_source_record_storage.marc_indexers_{%} ALTER COLUMN version SET NOT NULL;
        commit;
        raise info 'Constraint has been created on "{tenantId}_mod_source_record_storage.idx_marc_indexers_%_version" column', suffix;
    end loop;
end;
$$;

6. Create index marc_records_tracking table

Create index marc_records_tracking table

CREATE INDEX IF NOT EXISTS idx_marc_records_tracking_dirty ON {tenantId}_mod_source_record_storage.marc_records_tracking USING btree (is_dirty);

Where {tenantId} - the owner (tenant) id for which the script should be run.