

Reading and processing very large tables efficiently (millions of records)

 Language: [Deutsch](#) · [English](#)

Reading and processing very large tables efficiently (millions of records)

Process large tables efficiently – read in chunks and replace OFFSET with keyset pagination.

HOW-TO

DEVELOPER

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [Related pages](#)

Two basic rules to process very large tables (millions of records) efficiently.

1. Always fetch data **in chunks**.
2. When reading the whole table, **avoid OFFSET** – page forward via the last ID instead (keyset pagination).

```
final long CHUNKSIZE = 5000;

long lastID = 0L;

Map<UID, FieldMeta<?>> mpFields = metaProvider.getAllEntityFieldsByEntity(entity);
DbQueryBuilder builder = dataBaseHelper.getDbAccess().getQueryBuilder();

for (;;) {
    DbCondition cond = builder.plainCondition("t.INTID > " + lastID);

    CollectableSearchCondition csc = new CollectableDbCondition(cond);
    CollectableSearchExpression cse = new CollectableSearchExpression(csc);

    Collection<EntityObjectV0<Long>> lstEOs = entityObjectFacade.getEntityObjectsChunk(entity, mpFields.values(),
    null, cse, 0L, CHUNKSIZE - 1);

    if (lstEOs.isEmpty()) {
        return;
    }

    for (EntityObjectV0<Long> eo : lstEOs) {
        ....

        lastID = eo.getPrimaryKey();
    }
}
```

Related pages

Conventions for Nuclos API development

API.

[Open](#)

