TYPO3 Extbase

- Extbase - Datenbank Abfragen
  - Objekte sortiert durch Redakteur erhalten (e. g. selectMultipleSideBySide)

Extbase - Datenbank Abfragen

Objekte sortiert durch Redakteur erhalten (e. g. selectMultipleSideBySide)

Wenn man FlexForm oder TCA Feld wie zum Beispiel ein select -> selectMultipleSideBySide hat, und dieses nicht auf einer MM-Table basiert, funktioniert die Manuelle Item Sortierung nicht in der DB Abfrage in deiner Extension.

Es muss also manuell eine Möglichkeit geschaffen werden wie man der DB mitteilt in welcher Reihenfolge man die Ergebnisse haben möchte.

Nachfolgend ein Beispiel dazu:

```
// Raw QueryBuilder für die performante Sortierung
$queryBuilder = $this->connectionPool->getQueryBuilderForTable('tx_your_table');



// UID-Liste formatieren z. B.: $uidList = '969,318,32,139,384,600'
$uidList = implode(',', array_map('intval', $uids));



$queryBuilder
    ->select('*')
    ->from('tx_your_table')
    ->where(
        $queryBuilder->expr()->in('uid', $uids)
    );



$concreteQueryBuilder = $queryBuilder->getConcreteQueryBuilder();
$concreteQueryBuilder
    // proper quoting must be done manually on the concrete query builder
```

```
    ->orderBy('FIELD(' .$queryBuilder->quoteIdentifier('uid') . ',' . $uidList . ')');



$rows = $queryBuilder->executeQuery()->fetchAllAssociative();
```

Möchte man nun gar keinen $queryBuilder verwenden weil man eventuell direkt Objekte als Ergebnis haben möchte, kann man leider den TYPO3 Extbase eigenen $query nicht verwenden. Dazu geht man über nachfolgenden Umweg:
Man schreibt seine Query wie oben über den QueryBuilder und mappt die Ergebnisse dann zu einem Objekt:

```
// Konvertiere Rows zu Extbase Objects
// TravelList ist hierbei dein Domain/Model
$dataMapper =
GeneralUtility::makeInstance(\TYPO3\CMS\Extbase\Persistence\Generic\Mapper\DataMapper::class);
$objects = [];
foreach ($rows as $row) {
    $objects[] = $dataMapper->map(TravelList::class, [$row])[0];
}
```