# Entity Resolution On-Demand

# Appendices

## A QUERY GENERATION

The definition of the batches of queries for the experimental evaluation was carried out according to the following strategies.

For the AND queries on SIGMOD20, the selection is based on a random value among the ten most frequent brands and on a random model series associated with the selected brand; for the OR queries, on two random values chosen from the ten most frequent brands. For the AND queries on SIGMOD21 and Altosight, the selection is based on the list of all nine brands in the dataset and a list of the most common USB stick sizes; for the OR queries, on two random brands. Finally, for the AND queries on Funding, the selection is based on a list of the most common areas in which the organizations are active and on a frequent substring of their names (e.g., *association*, *inc*, etc.); for the OR queries, on two random areas.

The criteria are slightly different for the batches of AND queries used for *Batch-query-baseline*. For SIGMOD20, in this case we consider the brand and the average value of megapixels (which is required to be above a certain threshold), while for Altosight and SIGMOD21 we consider the brand and the average price (below a certain threshold). This choice was adopted since for these datasets, differently from what happens in Funding, the attributes used in standard batches do not present an intra-cluster value variance significant enough to show the potential performance degradation introduced by the adoption of the described *Batch-query-baseline* in a generic scenario.

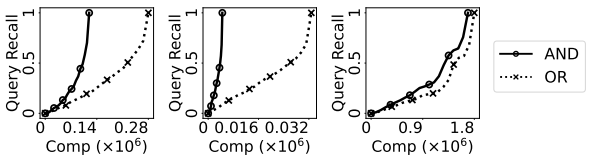
## B MANUALLY-DEVISED BLOCKING

In addition to unsupervised blocking, we also evaluated manually-devised blocking strategies. We report in Table 1 the characteristics of the best configuration we could find for each dataset. Yet, for Funding, the blocking strategy yields a low recall. However, it represents a plausible real-world scenario, which gives the opportunity to study whether *BrewER* is affected by an aggressive blocking strategy. As depicted in Figure 1, no significant differences in performance can be detected, apart from the number of comparisons depending on the precision of the blocking function. Details about the adopted blocking strategies are illustrated below.

For SIGMOD20, for each record two keys are extracted from brand and model attribute values as follows. The first key is extracted by removing punctuation marks and numeric characters from the two attributes and concatenating the remaining strings. The second key is generated with the numerical characters of model. Then, these two keys are employed to build an inverted index of the records: each key corresponds to a block and two records are indexed together in the same block if they share at least one key. For Altosight, for each record a key is generated by concatenating brand and size values (if both attribute values are different from null) and removing white spaces. The key is then used for creating the blocks as for SIGMOD20. Of course, the same criterion could be applied to its subset SIGMOD21. For Funding dataset, two keys are considered for each record. Firstly, if name attribute value is not null, up to two tokens before the first comma are extracted (this is done to remove organization suffixes); then, punctuation marks, numeric characters and white spaces are removed to yield the final key. A second key is extracted from address value (if not null) by removing punctuation marks, numeric characters and white spaces. As for the other datasets, these keys are employed for creating the blocks. It is important to notice that, differently from the blocks produced using JedAI, the ones obtained through these manually-devised blocking strategies are designed to be transitively closed; so, the inverted index phase is followed by the computation of the transitive closure of the produced blocks.

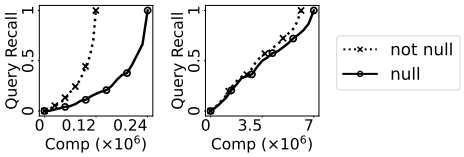
**Table 1: Manually-devised blocking characteristics.**

| Dataset | Recall | Precision | $F_1$ |
|---|---|---|---|
| SIGMOD20 | 0.993 | 0.014 | 0.028 |
| Altosight | 0.999 | 0.086 | 0.158 |
| Funding | 0.586 | 0.023 | 0.044 |




(a) SIGMOD20 (b) Altosight (c) Funding

**Figure 1: Progressive Recall with manually-devised blocking.**




(a) SIGMOD20 (BL) (b) Altosight

**Figure 2: Progressive Recall with Missing Values.**

## C PERFORMANCE WITH MISSING VALUES

With this experiment, we want to evaluate the impact of null values in the ordering attribute. We consider only SIGMOD20 (with blocking) and Altosight, since SIGMOD21 and Funding do not present null values in their ordering attribute. Thus, 16.2 thousand and 1.5 thousand of additional records with null values have been added to SIGMOD20 and Altosight, respectively, for this experiment. In SIGMOD20 the new records are scattered among around 6 thousand of entities, while in Altosight only on 30 entities.

The experiment results are reported in Figure 2. For Altosight (Figure 2b), the progressive recall of the queries executed on the dataset with null values has a curve similar in shape to the one generated without considering the null values. As a matter of fact, only 30 entities have records with null values, but none of them has a final resolved ordering value equal to null: their priorities in the priority queue do not change. The only difference is the slightly

higher number of comparisons needed for the additional records with `null` values.

Differently, Figure 2a showcases the negative effect of the high number of entities with `null` ordering value of `SIGMOD20`. Besides the higher number of comparisons needed for resolving the entire dataset (which has 16.2 thousand records more than the dataset without `null` values), the progressive recall curve is more flattened in presence of `null` values for the first circa 90% of the comparisons, and steeper at the end. This is due to the fact that many entities have an ordering value equal to `null` (i.e., all the ordering values of their records are `null`): they have the lowest priority in the priority queue, so they are compared, resolved and emitted only at the end of the processing.