

Code Summary: Extended Reference Pattern

The following pipeline creates an extended reference by updating all documents in the `reviews` collection to include relevant product information from the `books` collection using the `$lookup` stage.

```
var reshape_review_docs_pipeline = [
  {
    $lookup: {
      from: "books",
      localField: "product_id",
      foreignField: "product_id",
      as: "product_info",
    },
  },
  {
    $unwind: {
      path: "$product_info",
      includeArrayIndex: "string",
      preserveNullAndEmptyArrays: false,
    },
  },
  {
    $project: {
      _id: "$_id",
      "product.product_id": "$product_id",
      "product.product_type":
        "$product_info.product_type",
      "product.title": "$product_info.title",
      "review.user_id": "$user_id",
      "review.reviewTitle": "$reviewTitle",
      "review.reviewBody": "$reviewBody",
      "review.date": "$date",
      "review.stars": "$stars",
    },
  },
  {
    $merge: {
      into: "reviews",
```

```

    on: "_id",
    whenMatched: "replace",
    whenNotMatched: "discard",
  },
},
]

```

```
db.reviews.aggregate(reshape_review_docs_pipeline)
```

Once our new document with the extended reference is formed we use the

`$merge`

stage to merge the output into the existing reviews collection.

```

...
{
  $merge: {
    into: "reviews",
    on: "_id",
    whenMatched: "replace",
    whenNotMatched: "discard",
  },
},
...

```

Finally, we run the aggregation pipeline by invoking the aggregate method on the reviews collection, passing in the pipeline variable.

```
db.reviews.aggregate(reshape_review_docs_pipeline)
```