MONGODB AGGREGATE

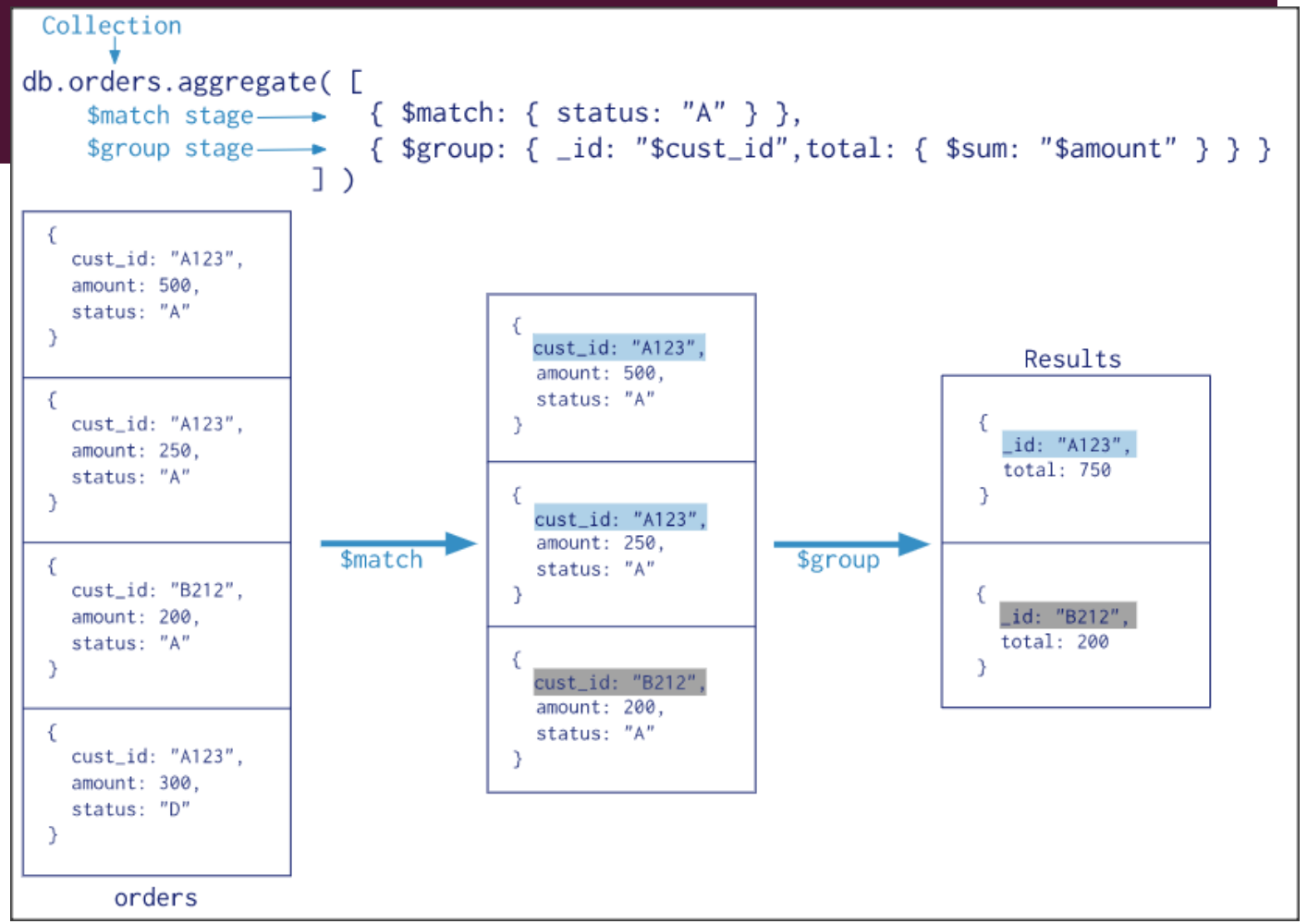
RISMAYANA@POLTEKTEDC.AC.ID

WHAT IS AGGREGATION

- Aggregations operations process data records and return computed results.
- Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result.
- In SQL count(*) and with group by is an equivalent of mongodb aggregation.

- The aggregate() Method
  - For the aggregation in MongoDB, you should use aggregate() method.
- Syntax
  - Basic syntax of aggregate() method is as follows
  - >db.COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)

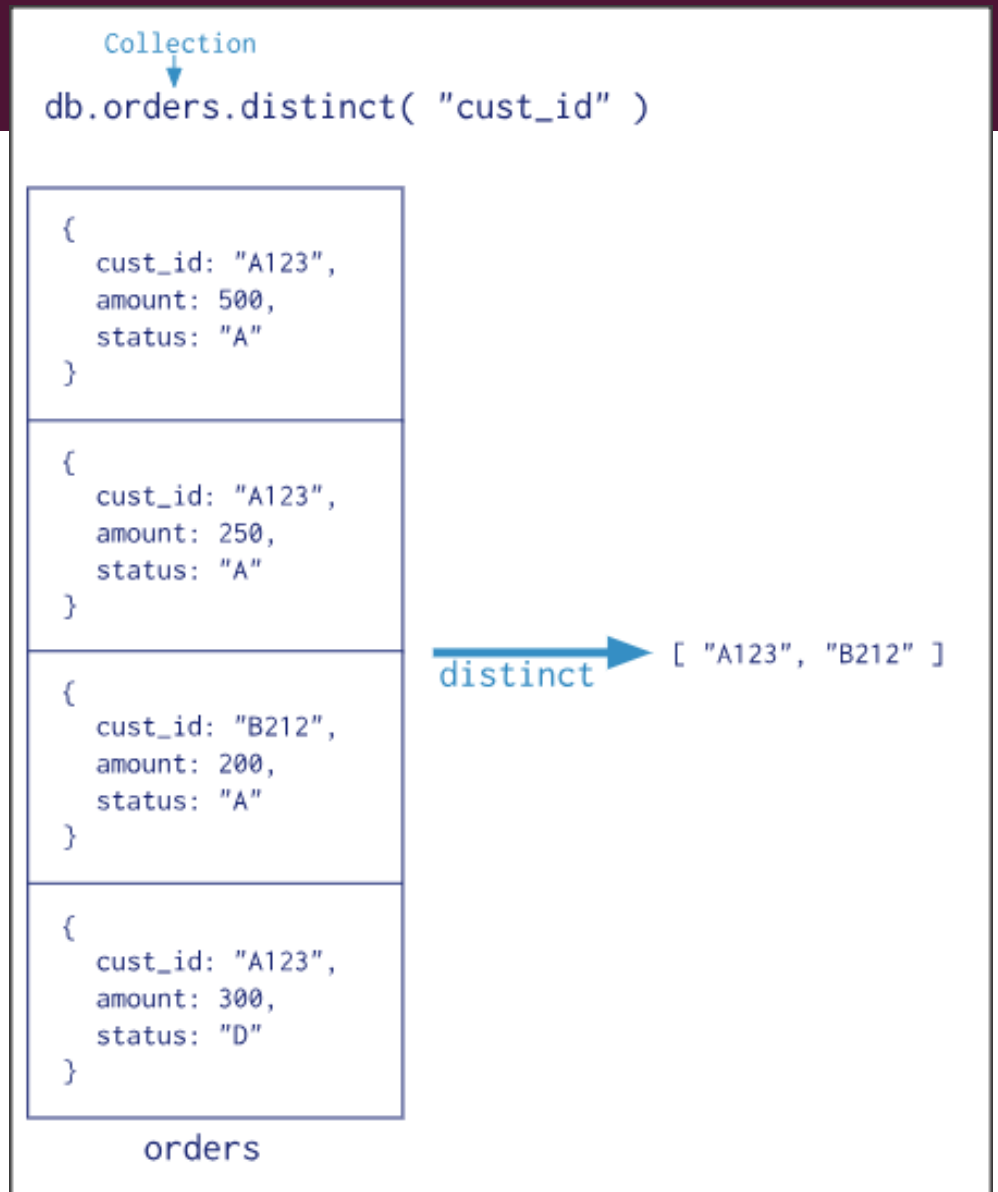
- **$project** − Used to select some specific fields from a collection.
- **$match** − This is a filtering operation and thus this can reduce the amount of documents that are given as input to the next stage.
- **$group** − This does the actual aggregation as discussed above.
- **$sort** − Sorts the documents.
- **$skip** − With this, it is possible to skip forward in the list of documents for a given amount of documents.
- **$limit** − This limits the amount of documents to look at, by the given number starting from the current positions.



Expression	Description	Example
$sum	Sums up the defined value from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$sum : "$likes"}}}])
$avg	Calculates the average of all given values from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$avg : "$likes"}}}])
$min	Gets the minimum of the corresponding values from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$min : "$likes"}}}])
$max	Gets the maximum of the corresponding values from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$max : "$likes"}}}])

Expression	Description	Example
$push	Inserts the value to an array in the resulting document.	db.mycol.aggregate([{$group : {_id : "$by_user", url : {$push: "$url"}}}])
$addToSet	Inserts the value to an array in the resulting document but does not create duplicates.	db.mycol.aggregate([{$group : {_id : "$by_user", url : {$addToSet : "$url"}}}])
$first	Gets the first document from the source documents according to the grouping. Typically this makes only sense together with some previously applied “$sort”-stage.	db.mycol.aggregate([{$group : {_id : "$by_user", first_url : {$first : "$url"}}}])
$last	Gets the last document from the source documents according to the grouping. Typically this makes only sense together with some previously applied “$sort”-stage.	db.mycol.aggregate([{$group : {_id : "$by_user", last_url : {$last : "$url"}}}])

- The SELECT DISTINCT statement is used to return only distinct (different) values.
- Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.



- Munculkan nama customer dan jumlah total yang telah dibayarkan selama ini!
- Munculkan 5 orang yang melakukan peminjaman paling banyak
- Munculkan film yang terdapat pada collection actor kemudian hitung jumlahnya
- Munculkan genre file yang terdapat pada collection payment kemudian hitung jumlahnya
- Munculkan customer yang berasal dari US
- Berapakah jumlah peminjaman yang terjadi pada tanggal 17 agustus 2005 yang berasal dari US