

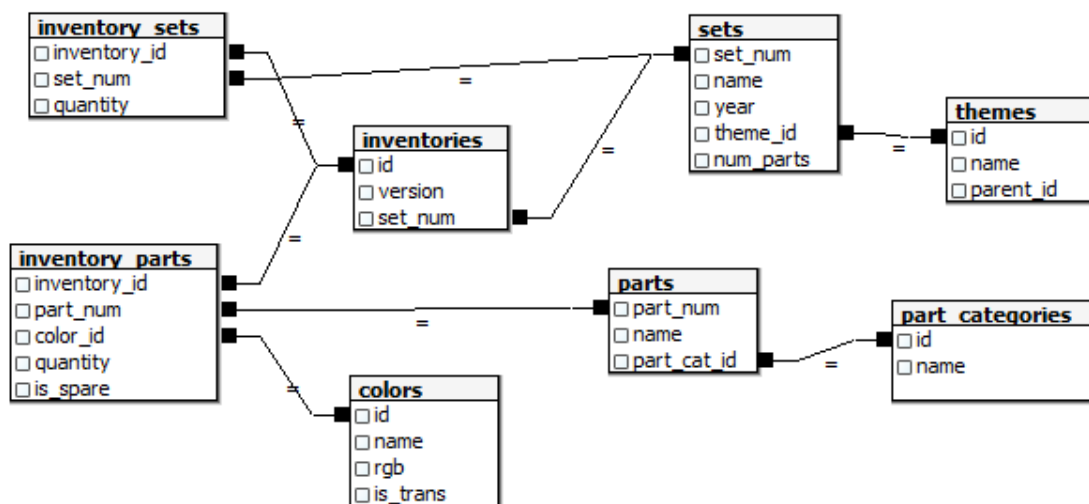
Proyecto “LEGO INC”

Objetivo:

De una base de datos proporcionada por la compañía LEGO se desea saber el estatus de la misma en cuanto a tema de inventarios, además de hacer un análisis de datos, se debe proporcionar una posible causa del problema derivado de dicho análisis, así como una solución.

Bases de Datos:

La información está separada en 8 tablas organizadas bajo el esquema que se muestra a continuación.



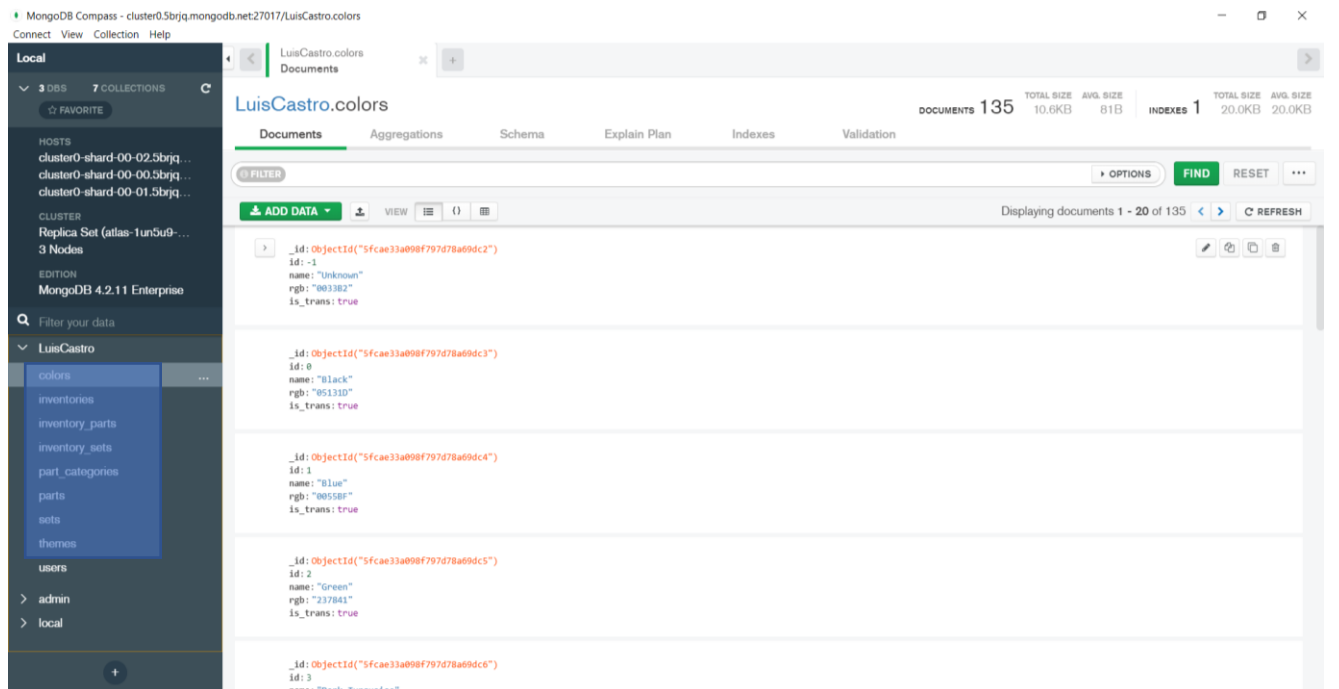
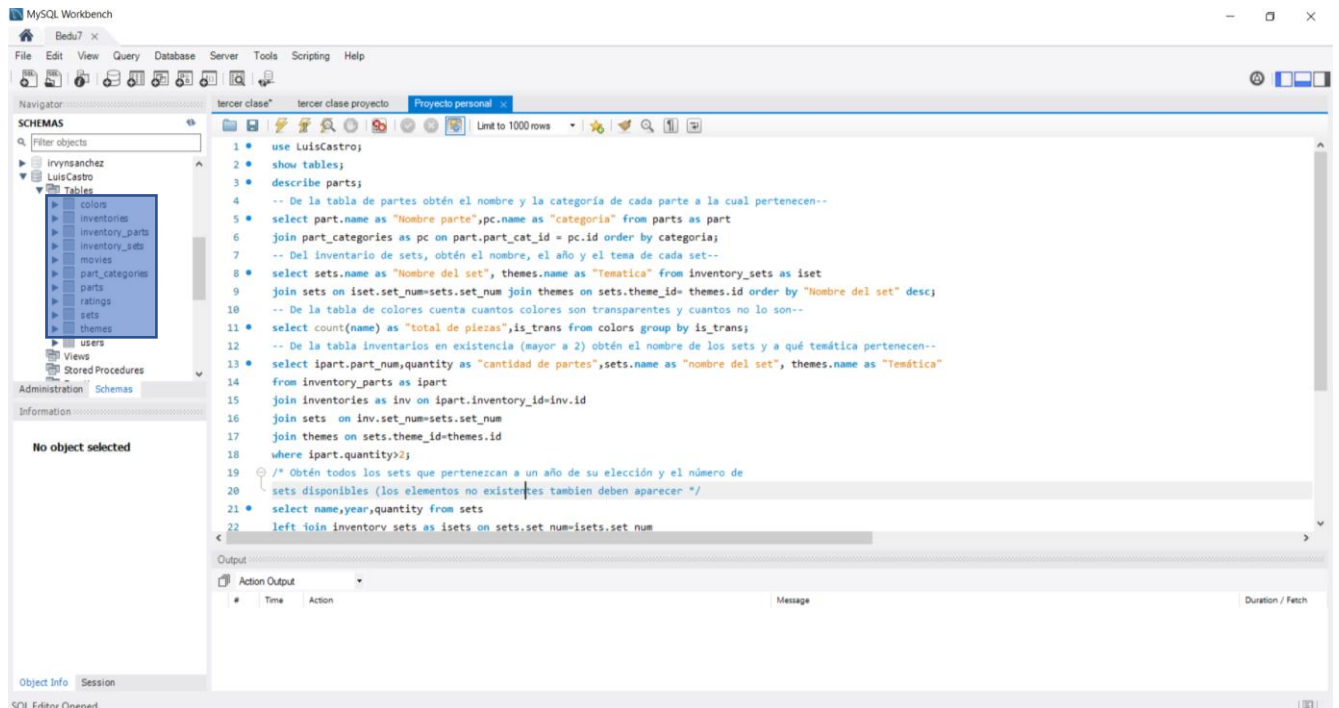
A continuación, se muestra el contenido de cada campo.

- **COLORS**
 - Id
 - Unique ID for this color.
 - Name
 - The human-readable name of the color.
 - Rgb
 - The approximate RGB color.
 - is_trans
 - Whether or not the given color is transparent/translucent.
- **INVENTORIES**
 - Id
 - Unique ID for this inventory entry.
 - Versión
 - Version number.
 - set_num
 - Set number (form `sets.csv`).
- **INVENTORY_PARTS**

- inventory_id
 - Unique ID for the inventory this part is appearing in. This is the same as the id value in `inventories.csv`.
- part_num
 - Unique ID for the part.
- color_id
 - Unique ID for the color, as per `colors.csv`.
- Quantity
 - The number of copies of this part included in the set!
- is_spare
 - Whether or not this is a spare part. Spare parts are additional parts not needed to finish the set.
- INVENTORY_SETS
 - inventory_id
 - Unique inventory ID from `inventories.csv`.
 - set_num
 - Unique set ID from `sets.csv`.
 - quantity
 - The quantity of the inventory included.
- PART_CATEGORIES
 - id
 - Unique ID for the part category.
 - name
 - The category of stuff the part is in.
- Parts
 - part_num
 - Unique ID for the part.
 - name
 - Name of the part
 - part_cat_id
 - Part category unique ID (from `part_categories.csv`).
- Sets
 - Set_num
 - Unique set ID.
 - Name
 - The name of the set.
 - Year
 - Year the set was published.
 - Theme_id
 - Unique ID for the theme used for the set (from `themes.csv`).
 - Num_parts
 - The number of parts included in the set.
- Themes
 - Id
 - Theme unique ID.
 - Name
 - Name of the theme.
 - Parent_id
 - Unique ID for the larger theme, if there is one.

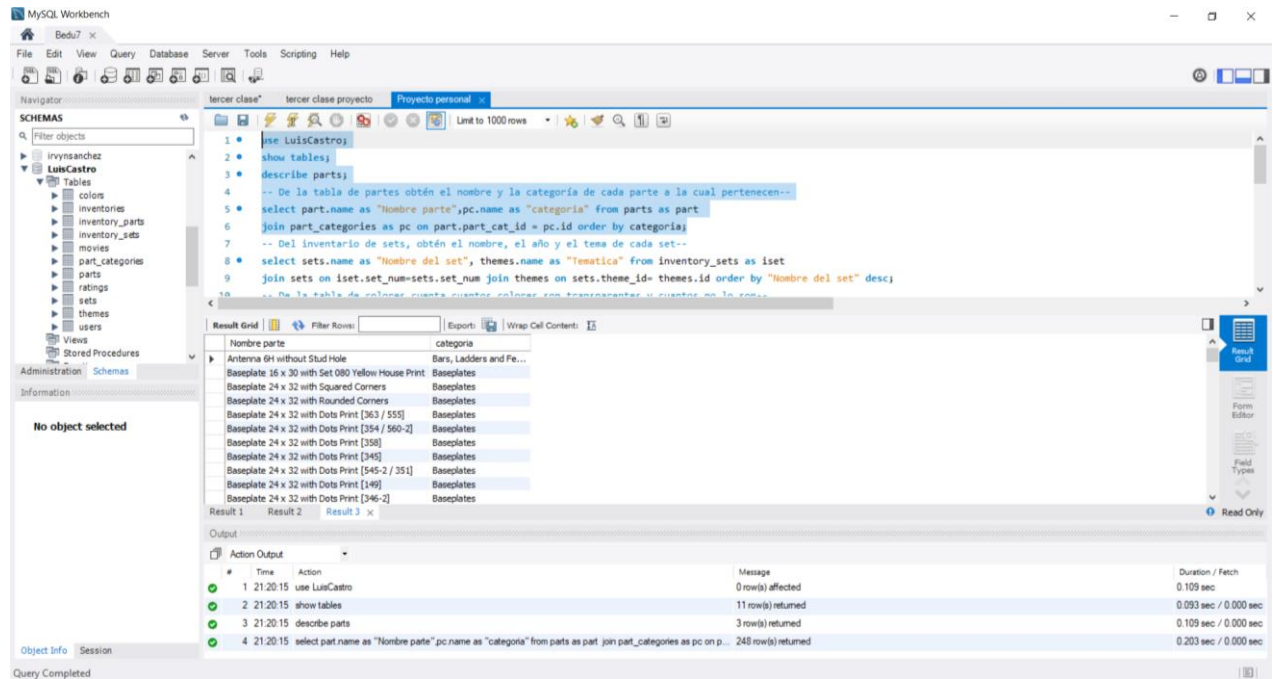
Creación de Bases de datos en MYSQL y MongoDB

Por motivos de rendimiento y tiempo la base de datos que fue ingresada a MYSQL fue reducida en cantidad de datos, esto puede llegar a causar discrepancias en las consultas.



Consultas

1.-De la tabla de partes obtén el nombre y la categoría de cada parte a la cual pertenecen



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use LuisCastro;
2 show tables;
3 describe parts;
4 -- De la tabla de partes obtén el nombre y la categoría de cada parte a la cual pertenecen--
5 select part.name as "Nombre parte", pc.name as "Categoría" from parts as part
6 join part_categories as pc on part.part_cat_id = pc.id order by categoría;
7 -- Del inventario de sets, obtén el nombre, el año y el tema de cada set--
8 select sets.name as "Nombre del set", themes.name as "Temática" from inventory_sets as iset
9 join sets on iset.set_num=sets.set_num join themes on sets.theme_id= themes.id order by "Nombre del set" desc;
10 -- De la tabla de colores, cuenta cuantos colores son transparentes y cuantos no lo son--
```

The Results window shows the output of the first query:

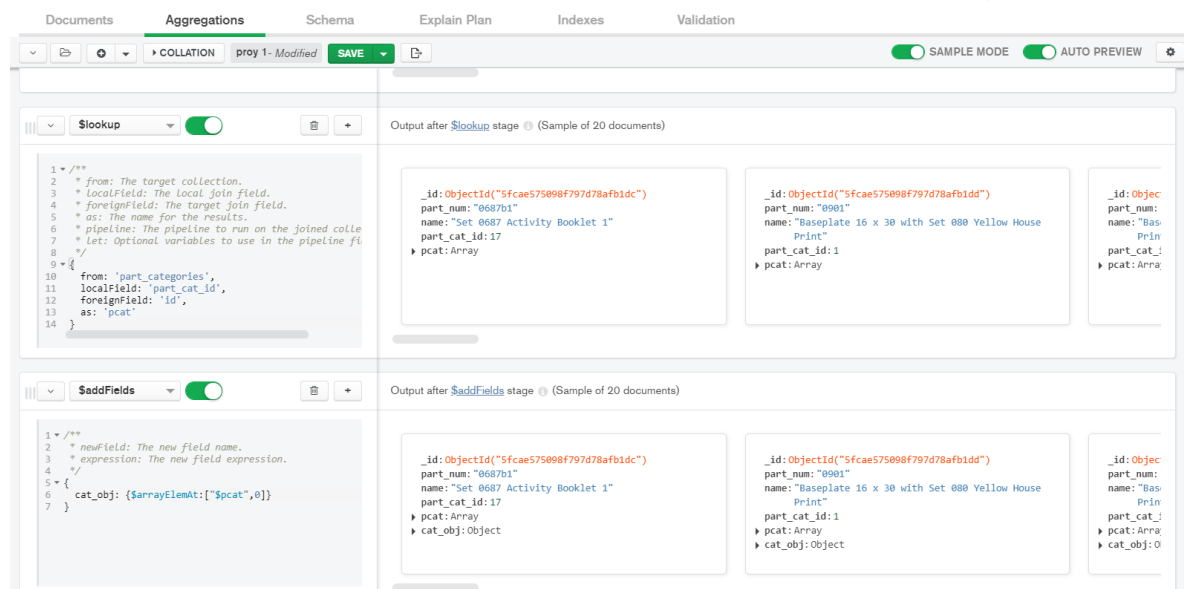
Nombre parte	Categoría
Antenna 6H without Stud Hole	Bars, Ladders and Fe...
Baseplate 16 x 30 with Set 080 Yellow House Print	Baseplates
Baseplate 24 x 32 with Squared Corners	Baseplates
Baseplate 24 x 32 with Rounded Corners	Baseplates
Baseplate 24 x 32 with Dots Print [363 / 555]	Baseplates
Baseplate 24 x 32 with Dots Print [354 / 560-2]	Baseplates
Baseplate 24 x 32 with Dots Print [358]	Baseplates
Baseplate 24 x 32 with Dots Print [345]	Baseplates
Baseplate 24 x 32 with Dots Print [545-2 / 351]	Baseplates
Baseplate 24 x 32 with Dots Print [149]	Baseplates
Baseplate 24 x 32 with Dots Print [346-2]	Baseplates

The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	21:20:15	use LuisCastro	0 row(s) affected	0.109 sec
2	21:20:15	show tables	11 row(s) returned	0.093 sec / 0.000 sec
3	21:20:15	describe parts	3 row(s) returned	0.109 sec / 0.000 sec
4	21:20:15	select part name as "Nombre parte", pc name as "Categoría" from parts as part join part_categories as pc on p...	248 row(s) returned	0.203 sec / 0.000 sec

LuisCastro.parts

DOCUMENTS 26.0k TOTAL SIZE 3.2MB AVG. SIZE 129B INDEXES 1 TOTAL SIZE 268.0KB AVG. SIZE 268.0KB



The screenshot shows the MongoDB Compass interface for the LuisCastro.parts collection. The Aggregations tab is active, showing a pipeline with two stages: \$lookup and \$addFields.

\$lookup stage:

```
1 /**
2  * from: The target collection.
3  * localField: The local join field.
4  * foreignField: The target join field.
5  * as: The name for the results.
6  * pipeline: The pipeline to run on the joined colle
7  * let: Optional variables to use in the pipeline fu
8  */
9 {
10   from: 'part_categories',
11   localField: 'part_cat_id',
12   foreignField: 'id',
13   as: 'pcat'
14 }
```

\$addFields stage:

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   cat_obj: {$arrayElemAt: ['$pcat', 0]}
7 }
```

The Output window shows the results of the aggregation pipeline, displaying the joined data for the first three documents.

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: proy 1- Modified SAVE SAMPLE MODE AUTO PREVIEW

\$addFields Output after \$addFields stage (Sample of 20 documents)

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   categoria: "$cat_obj.name"
7 }
```

Output after \$project stage (Sample of 20 documents)

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   part_num:1,
7   _id:0,
8   name:1,
9   categoria:1
10 }
```

Output after \$addFields stage (Sample of 20 documents)

```
{
  "_id": "ObjectId('5fcae575098f797d78afb1dd')",
  "part_num": "0687b1",
  "name": "Set 0687 Activity Booklet 1",
  "part_cat_id": 17,
  "pcat": Array,
  "cat_obj": Object,
  "categoria": "Non-LEGO"
}
```

Output after \$project stage (Sample of 20 documents)

```
{
  "part_num": "0687b1",
  "name": "Set 0687 Activity Booklet 1",
  "categoria": "Non-LEGO"
}
```

Output after \$addFields stage (Sample of 20 documents)

```
{
  "_id": "ObjectId('5fcae575098f797d78afb1dd')",
  "part_num": "0901",
  "name": "Baseplate 16 x 30 with Set 080 Yellow House",
  "Print": "Print",
  "part_cat_id": 1,
  "pcat": Array,
  "cat_obj": Object,
  "categoria": "Baseplates"
}
```

Output after \$project stage (Sample of 20 documents)

```
{
  "part_num": "0901",
  "name": "Baseplate 16 x 30 with Set 080 Yellow House",
  "categoria": "Baseplates"
}
```

Output after \$addFields stage (Sample of 20 documents)

```
{
  "_id": "ObjectId('5fcae575098f797d78afb1dd')",
  "part_num": "0901",
  "name": "Baseplate 16 x 30 with Set 080 Yellow House",
  "Print": "Print",
  "part_cat_id": 1,
  "pcat": Array,
  "cat_obj": Object,
  "categoria": "Baseplates"
}
```

Output after \$project stage (Sample of 20 documents)

```
{
  "part_num": "0901",
  "name": "Baseplate 16 x 30 with Set 080 Yellow House",
  "categoria": "Baseplates"
}
```

2.-Del inventario de sets, obtén el nombre, el año y el tema de cada set

MySQL Workbench

Schema: LuisCastro

Query:

```
4 -- De la tabla de partes obtén el nombre y la categoría de cada parte a la cual pertenecen--
5 select part.name as "Nombre parte",pc.name as "categoria" from parts as part
6 join part_categories as pc on part.part_cat_id = pc.id order by categoria;
7
8 -- Del inventario de sets, obtén el nombre, el año y el tema de cada set--
9 select sets.name as "Nombre del set", themes.name as "tematica" from inventory_sets as iset
10 join sets on iset.set_num=sets.set_num join themes on sets.theme_id= themes.id order by "Nombre del set" desc;
11
12 -- De la tabla de colores cuenta cuantos colores son transparentes y cuantos no lo son--
13 select count(name) as "total de piezas",is_trans from colors group by is_trans;
14
15 -- De la tabla inventarios en existencia (mayor a 2) obtén el nombre de los sets y a qué temática pertenecen--
16 select iset.part_num,quantity as "cantidad de piezas",cat.name as "Nombre del cat", theme.name as "tematica"
17 from inventory_sets as iset join parts as part join part_categories as pc on iset.set_num=part.part_cat_id=pc.id
18 where iset.quantity > 2 order by iset.set_num;
```

Result Grid:

Nombre del set	Tematica
Horizon Express	RC Train
164 Action Set with Stickers	Hockey
TIE Fighter Collection	Star Wars Episode 4/5/6
Weetabix Promotional House 1	Building
Weetabix Promotional House 2	Building
Weetabix Promotional Windmill	Building
Birthday Pack Heart	Clkits
Birthday Pack Daisy	Clkits
Birthday Pack Star	Clkits
High Speed Train Car	World City

Result 4

Action Output

#	Time	Action	Message	Duration / Fetch
2	21:20:15	show tables	11 row(s) returned	0.093 sec / 0.000 sec
3	21:20:15	describe parts	3 row(s) returned	0.109 sec / 0.000 sec
4	21:20:15	select part name as "Nombre parte",pc name as "categoria" from parts as part join part_categories as pc on...	248 row(s) returned	0.203 sec / 0.000 sec
5	22:25:19	select sets name as "Nombre del set", themes name as "tematica" from inventory_sets as iset join sets on i...	10 row(s) returned	0.110 sec / 0.000 sec

Query Completed

LuisCastro.parts LuisCastro.inventory_sets Aggregations

LuisCastro.inventory_sets documents 2.8k TOTAL SIZE 206.0KB AVG. SIZE 74B INDEXES 1 TOTAL SIZE 52.0KB AVG. SIZE 52.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proxy 2 SAVE SAMPLE MODE AUTO PREVIEW

Stoosup Output after Stoosup stage (Sample of 20 documents)

```
1 /*
2 * from: The target collection.
3 * localField: The local join field.
4 * foreignField: The target join field.
5 * as: The name for the results.
6 * pipeline: The pipeline to run on the joined colle
7 * let: Optional variables to use in the pipeline f
8 */
9 {
10   from: 'sets',
11   localField: 'set_num',
12   foreignField: 'set_num',
13   as: 'sets'
14 }
```

Output after Stoosup stage (Sample of 20 documents)

```
{
  "_id": "ObjectID('5fcae538090f797d78afad85')",
  "inventory_id": 35,
  "set_num": "75911-1",
  "quantity": 1,
  "sets": Array
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad86')",
  "inventory_id": 35,
  "set_num": "75912-1",
  "quantity": 1,
  "sets": Array
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad87')",
  "inventory_id": 35,
  "set_num": "75913-1",
  "quantity": 1,
  "sets": Array
}
```

Stoosup Output after Stoosup stage (Sample of 20 documents)

```
1 /*
2 * from: The target collection.
3 * localField: The local join field.
4 * foreignField: The target join field.
5 * as: The name for the results.
6 * pipeline: The pipeline to run on the joined colle
7 * let: Optional variables to use in the pipeline f
8 */
9 {
10   from: 'themes',
11   localField: 'sets.theme_id',
12   foreignField: 'id',
13   as: 'theme'
14 }
```

Output after Stoosup stage (Sample of 20 documents)

```
{
  "_id": "ObjectID('5fcae538090f797d78afad85')",
  "inventory_id": 35,
  "set_num": "75911-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad86')",
  "inventory_id": 35,
  "set_num": "75912-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad87')",
  "inventory_id": 35,
  "set_num": "75913-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array
}
```

LuisCastro.inventory_sets documents 2.8k TOTAL SIZE 206.0KB AVG. SIZE 74B INDEXES 1 TOTAL SIZE 52.0KB AVG. SIZE 52.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proxy 2 SAVE SAMPLE MODE AUTO PREVIEW

\$addFields Output after \$addFields stage (Sample of 20 documents)

```
1 /*
2 * newField: The new field name.
3 * expression: The new field expression.
4 */
5 {
6   objset: { $arrayElemAt: [ '$sets', 0 ] },
7   objtem: { $arrayElemAt: [ '$theme', 0 ] }
8 }
```

Output after \$addFields stage (Sample of 20 documents)

```
{
  "_id": "ObjectID('5fcae538090f797d78afad85')",
  "inventory_id": 35,
  "set_num": "75911-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array,
  "objset": Object,
  "objtem": Object
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad86')",
  "inventory_id": 35,
  "set_num": "75912-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array,
  "objset": Object,
  "objtem": Object
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad87')",
  "inventory_id": 35,
  "set_num": "75913-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array,
  "objset": Object,
  "objtem": Object
}
```

\$addFields Output after \$addFields stage (Sample of 20 documents)

```
1 /*
2 * newField: The new field name.
3 * expression: The new field expression.
4 */
5 {
6   nombre_set: "$objset.name",
7   año: "$objset.year",
8   tematica: "$objtem.name"
9 }
```

Output after \$addFields stage (Sample of 20 documents)

```
{
  "set_num": "75911-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array,
  "objset": Object,
  "objtem": Object,
  "nombre_set": "McLaren Mercedes Pit Stop",
  "año": 2015,
  "tematica": "Speed Champions"
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad86')",
  "inventory_id": 35,
  "set_num": "75912-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array,
  "objset": Object,
  "objtem": Object,
  "nombre_set": "Borracho, Bitch, Et. Efnich. I fna",
  "año": 2015,
  "tematica": "Speed Champions"
}
```

```
{
  "_id": "ObjectID('5fcae538090f797d78afad87')",
  "inventory_id": 35,
  "set_num": "75913-1",
  "quantity": 1,
  "sets": Array,
  "theme": Array,
  "objset": Object,
  "objtem": Object,
  "nombre_set": "Borracho, Bitch, Et. Efnich. I fna",
  "año": 2015,
  "tematica": "Speed Champions"
}
```

\$project Output after \$project stage (Sample of 20 documents)

```
1 /*
2 * from: The target collection.
3 * localField: The local join field.
4 * foreignField: The target join field.
5 * as: The name for the results.
6 * pipeline: The pipeline to run on the joined colle
7 * let: Optional variables to use in the pipeline f
8 */
9 {
10   from: 'sets',
11   localField: 'set_num',
12   foreignField: 'set_num',
13   as: 'sets'
14 }
```

LuisCastro.inventory_sets

DOCUMENTS 2.8K TOTAL SIZE 206.8KB AVG. SIZE 74B INDEXES 1 TOTAL SIZE 52.0KB AVG. SIZE 52.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: proy 2 SAVE SAMPLE MODE AUTO PREVIEW

\$addFields Output after \$addFields stage (Sample of 20 documents)

```

1 // **
2 * newField: The new field name.
3 * expression: The new field expression.
4 */
5 {
6   nombre_set: "$objset.name",
7   año: "$objset.year",
8   tematica: "$objset.name"
9 }

```

set_num: 75911-1
quantity: 1
sets: Array
theme: Array
objset: Object
nombre_set: "McLaren Mercedes Pit Stop"
año: 2015
tematica: "Speed Champions"

_id: ObjectId("5fcae51889ef797d78afa686")
inventory_id: 35
set_num: "75912-1"
quantity: 1
sets: Array
theme: Array
objset: Object
nombre_set: "Porsche 911 GT3 RS"
año: 2015
tematica: "Speed Champions"

_id: ObjectId("5fcae51889ef797d78afa686")
inventory_id: 35
set_num: "75912-1"
quantity: 1
sets: Array
theme: Array
objset: Object
nombre_set: "Porsche 911 GT3 RS"
año: 2015
tematica: "Speed Champions"

\$project Output after \$project stage (Sample of 20 documents)

```

1 // **
2 * specifications: The fields to
3 * include or exclude.
4 */
5 {
6   nombre_set,
7   año,
8   tematica,
9   _id: 0
10 }

```

nombre_set: "McLaren Mercedes Pit Stop"
año: 2015
tematica: "Speed Champions"

nombre_set: "Porsche 911 GT3 RS"
año: 2015
tematica: "Speed Champions"

nombre_set: "Porsche 911 GT3 RS"
año: 2015
tematica: "Speed Champions"

ADD STAGE

3.- De la tabla de colores cuenta cuantos colores son transparentes y cuantos no lo son

Limit to 1000 rows

```

4 -- De la tabla de partes obtén el nombre y la categoría de cada parte a la cual pertenecen--
5 • select part.name as "Nombre parte",pc.name as "categoría" from parts as part
6 join part_categories as pc on part.part_cat_id = pc.id order by categoría;
7 -- Del inventario de sets, obtén el nombre, el año y el tema de cada set--
8 • select sets.name as "Nombre del set", themes.name as "Temática" from inventory_sets as iset
9 join sets on iset.set_num=sets.set_num join themes on sets.theme_id= themes.id order by "Nombre del set" desc;
10 -- De la tabla de colores cuenta cuantos colores son transparentes y cuantos no lo son--
11 • select count(name) as "total de piezas",is_trans from colors group by is_trans;
12 -- De la tabla inventarios en existencia (mayor a 2) obtén el nombre de los sets y a qué temática pertenecen--
13 • select iset.part_num,quantity as "cantidad de partes", sets.name as "nombre del set", themes.name as "Temática"

```

Result Grid Filter Rows: Export: Wrap Cell Contents

total de piezas	is_trans
107	f
28	t

Result 5

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled - Modified SAVE SAMPLE MODE AUTO PREVIEW

135 Documents in the Collection

Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

Output after \$group stage (Sample of 2 documents)

```
1 /*
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$is_trans",
7   verd: {
8     $sum: 1
9   }
10 }
```

Sample documents:

- `{ "_id": "f", "verd": 107 }`
- `{ "_id": "t", "verd": 28 }`

4.- De la tabla inventarios en existencia (mayor a 2) obtén el nombre de los sets y a qué temática pertenecen

Limit to 1000 rows

```
10 -- De la tabla de colores cuenta cuantos colores son transparentes y cuantos no lo son--
11 select count(name) as "total de piezas",is_trans from colors group by is_trans;
12 -- De la tabla inventarios en existencia (mayor a 2) obtén el nombre de los sets y a qué temática pertenecen--
13 select ipart.part_num,quantity as "cantidad de partes",sets.name as "nombre del set", themes.name as "Temática"
14 from inventory_parts as ipart
15 join inventories as inv on ipart.inventory_id=inv.id
16 join sets on inv.set_num=sets.set_num
17 join themes on sets.theme_id=themes.id
18 where ipart.quantity>2;
19 /* Obtén todos los sets que pertenecan a un año de su elección y el número de
```

Result Grid

part_num	cantidad de partes	nombre del set	Temática
4286	3	Mr. Bunny	Easter

Result 6


```

1 /**
2  * query: The query in MQL.
3  */
4 {
5   quantity:{$gt:2}
6 }

```

_id:ObjectId("5fcae538098f797d78afa6d4")
inventory_id: 309
set_num: "4515-1"
quantity: 7

_id:ObjectId("5fcae538098f797d78afa6d5")
inventory_id: 309
set_num: "4520-1"
quantity: 5

\$lookup

Output after \$lookup stage (Sample of 20 documents)

```

1 /**
2  * from: The target collection.
3  * localField: The local join field.
4  * foreignField: The target join field.
5  * as: The name for the results.
6  * pipeline: The pipeline to run on the joined colle
7  * let: Optional variables to use in the pipeline fi
8  */
9 {
10   from: 'sets',
11   localField: 'set_num',
12   foreignField: 'set_num',
13   as: 'sets'
14 }

```

_id:ObjectId("5fcae538098f797d78afa6d4")
inventory_id: 309
set_num: "4515-1"
quantity: 7
sets: Array

_id:ObjectId("5fcae538098f797d78afa6d5")
inventory_id: 309
set_num: "4520-1"
quantity: 5
sets: Array

LuisCastro.inventory_sets

DOCUMENTS 2.8k

TOTAL SIZE 206.8KB

AVG. SIZE 74B

INDEXES 1

TOTAL SIZE 40.0KB

AVG. SIZE 40.0KB

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

COLLATION

Untitled- Modified

SAVE

SAMPLE MODE

AUTO PREVIEW

\$lookup

Output after \$lookup stage (Sample of 20 documents)

```

1 /**
2  * from: The target collection.
3  * localField: The local join field.
4  * foreignField: The target join field.
5  * as: The name for the results.
6  * pipeline: The pipeline to run on the joined colle
7  * let: Optional variables to use in the pipeline fi
8  */
9 {
10   from: 'themes',
11   localField: 'sets.theme_id',
12   foreignField: 'id',
13   as: 'tematica'
14 }

```

_id:ObjectId("5fcae538098f797d78afa6d4")
inventory_id: 309
set_num: "4515-1"
quantity: 7
sets: Array
tematica: Array

_id:ObjectId("5fcae538098f797d78afa6d5")
inventory_id: 309
set_num: "4520-1"
quantity: 5
sets: Array
tematica: Array

_id:ObjectId("5fcae538098f797d78afa6d6")
inventory_id: 309
set_num: "4525-1"
quantity: 3
sets: Array
tematica: Array

\$project

Output after \$project stage (Sample of 20 documents)

```

1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   sets:{name:1},
7   tematica:{name:1}
8 }

```

_id:ObjectId("5fcae538098f797d78afa6d4")
sets: Array
0: Object
name: "Straight Rails"
tematica: Array
0: Object
name: "9V"

_id:ObjectId("5fcae538098f797d78afa6d5")
sets: Array
tematica: Array

_id:ObjectId("5fcae538098f797d78afa6d6")
sets: Array
tematica: Array

/*De la primer consulta, agrega el color a cada parte además */