

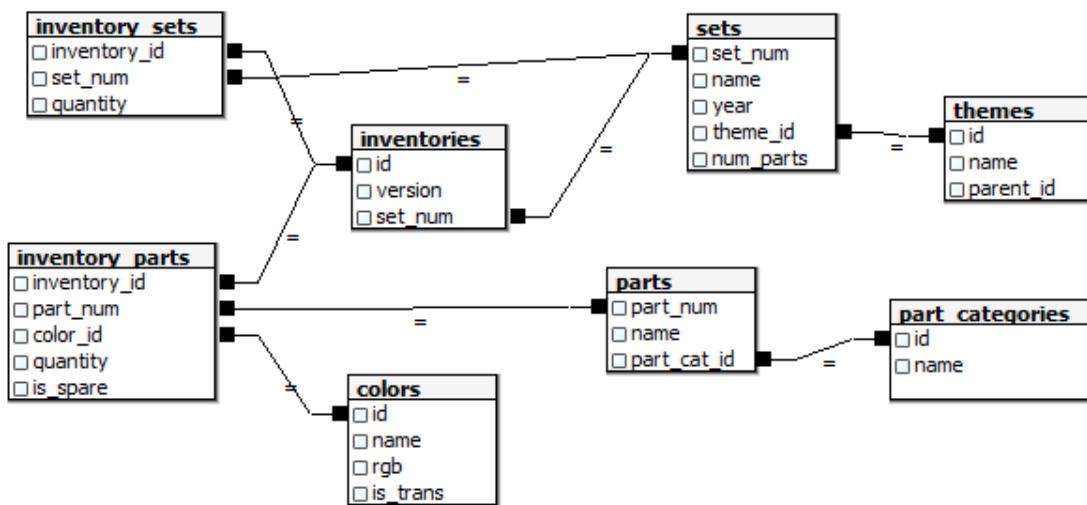
Proyecto “LEGO INC”

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.



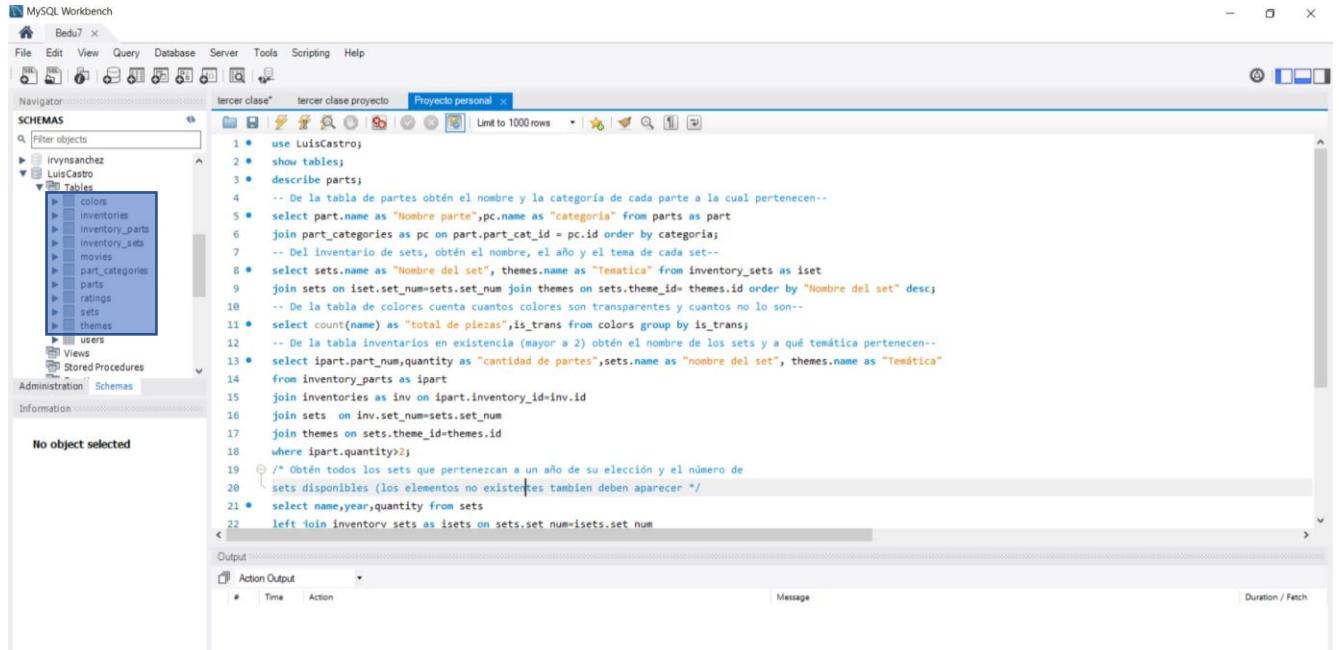
A continuación, se muestra el contenido de cada campo.

- 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 (from `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.



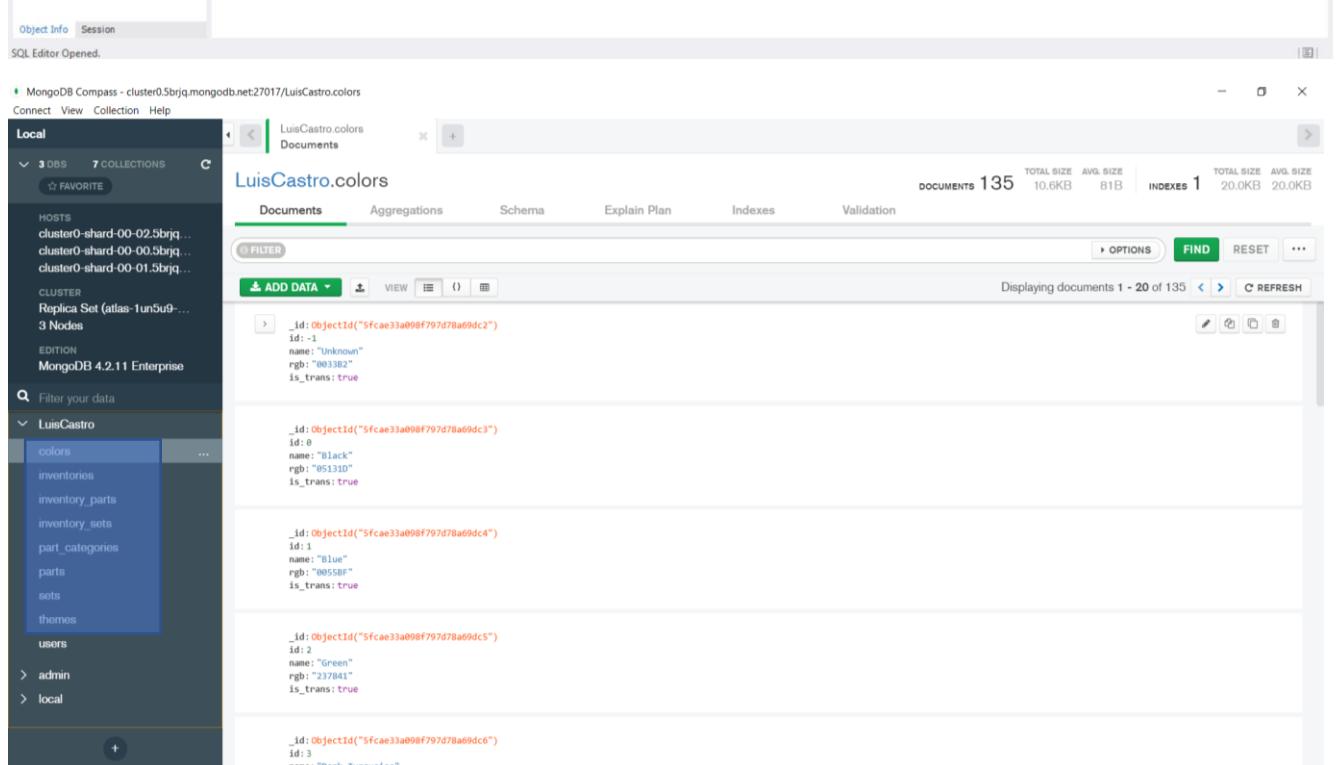
The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database 'LuisCastro' is selected. The left sidebar displays the schema structure under 'Schemas'. The main area contains a SQL editor window with the following query:

```

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 "categoria" from parts as part
join part_categories as pc on part.part_cat_id = pc.id order by categoria;
6
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 set
join sets on set.set_num=sets.set_num join themes on sets.theme_id=themes.id order by "Nombre del set" desc;
9
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"
from inventory_parts as ipart
join inventories as inv on ipart.inventory_id=inv.id
join sets on inv.set_num=sets.set_num
join themes on sets.theme_id=themes.id
where ipart.quantity>2;
14
15 /* Obtén todos los sets que pertenezcan a un año de su elección y el número de
16 sets disponibles (los elementos no existentes también deben aparecer */
17
18 • select name,year,quantity from sets
19
20 left join inventory_sets as isets on sets.set_num=isets.set_num
21
22

```

The 'Output' tab shows the results of the query execution.

The screenshot shows the MongoDB Compass interface. On the left, the 'Local' database is selected, showing 3 DBs and 7 collections. The 'colors' collection is currently selected. The main pane displays the 'Documents' tab for the 'LuisCastro.colors' collection, which contains 135 documents. The table header includes columns for DOCUMENTS, TOTAL SIZE, AVG. SIZE, and INDEXES. The document list shows entries like:

- `_id: ObjectId("5fcae33a090f797d78a69dc2")`
- `id: -1`
- `name: "Unknown"`
- `rgb: "#003B2B"`
- `is_trans: true`

Other entries follow a similar pattern with IDs ranging from 0 to 3, names like 'Black', 'Blue', 'Green', and 'Dark\_Turquoise', and various RGB values.

## Consultas

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

```

MySQL Workbench - LuisCastro
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas tercera clase* tercera clase proyecto Proyecto personal
Schemas
irvynsanchez
LuisCastro
Tables
colors
inventories
inventory_parts
inventory_sets
movies
part_categories
parts
ratings
sets
themes
Views
Stored Procedures
Administration Schemas Information
No object selected
Object Info Session
Query Completed

```

```

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 "Tematica" from inventory_sets as set
9 join sets on set.set_num=set.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.

```

Result Grid | Filter Rows | Export | Wrap Cell Content | Read Only

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

Action Output

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 129B INDEXES 1 TOTAL SIZE 268.0KB AVG. SIZE 268.0KB

Aggregations

Documents Aggregations Schema Explain Plan Indexes Validation 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 collection.
7 * let: Optional variables to use in the pipeline.
8 */
9 {
10   from: 'part_categories',
11   localField: 'part_cat_id',
12   foreignField: 'id',
13   as: 'pcat'
14 }

```

\_id:ObjectId("5fcae5750908f797d78afbd1dc")  
part\_num:"0687b1"  
name:"Set 0687 Activity Booklet 1"  
part\_cat\_id:17  
pcat:Array

\_id:ObjectId("5fcae5750908f797d78afbd1dc")  
part\_num:"0901"  
name:"Baseplate 16 x 30 with Set 080 Yellow House Print"  
part\_cat\_id:1  
pcat:Array

\_id:ObjectId("5fcae5750908f797d78afbd1dc")  
part\_num:"0901"  
name:"Baseplate 16 x 30 with Set 080 Yellow House Print"  
part\_cat\_id:1  
pcat:Array

\$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   $set_obj: {$arrayElemAt:[ "$pcat", 0 ]},
7 }

```

\_id:ObjectId("5fcae5750908f797d78afbd1dc")  
part\_num:"0687b1"  
name:"Set 0687 Activity Booklet 1"  
part\_cat\_id:17  
pcat:Array  
cat\_obj:Object

\_id:ObjectId("5fcae5750908f797d78afbd1dc")  
part\_num:"0901"  
name:"Baseplate 16 x 30 with Set 080 Yellow House Print"  
part\_cat\_id:1  
pcat:Array  
cat\_obj:Object

\_id:ObjectId("5fcae5750908f797d78afbd1dc")  
part\_num:"0901"  
name:"Baseplate 16 x 30 with Set 080 Yellow House Print"  
part\_cat\_id:1  
pcat:Array  
cat\_obj:Object

## Luis F Castro

**LuisCastro.parts**

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 }
```

**\_id: ObjectId("5fcfae575008f797d78afb1dc")**  
part\_num: "0687b1"  
name: "Set 0687 Activity Booklet 1"  
part\_cat\_id: 17  
pcat: Array  
cat\_obj: Object  
categoria: "Non-LEGO"

**\_id: ObjectId("5fcfae575008f797d78afb1dd")**  
part\_num: "0801"  
name: "Baseplate 16 x 30 with Set 080 Yellow House Print"  
part\_cat\_id: 1  
pcat: Array  
cat\_obj: Object  
categoria: "Baseplates"

**\_id: ObjectId("5fcfae575008f797d78afb1dc")**  
part\_num: "0801"  
name: "Baseplate 16 x 30 with Set 080 Yellow House Print"  
part\_cat\_id: 1  
pcat: Array  
cat\_obj: Object  
categoria: "Baseplates"

**\$project** 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 }
```

**part\_num: "0687b1"**  
name: "Set 0687 Activity Booklet 1"  
categoria: "Non-LEGO"

**part\_num: "0801"**  
name: "Baseplate 16 x 30 with Set 080 Yellow House Print"  
categoria: "Baseplates"

**part\_num: "0801"**  
name: "Baseplate 16 x 30 with Set 080 Yellow House Print"  
categoria: "Baseplates"

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

MySQL Workbench Beduz x

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas tercera clase proyecto Projeto personal

SCHEMAS LuisCastro

Tables colors inventories inventory\_parts inventory\_sets movies parts part\_categories ratings sets themes users Views Stored Procedures Administration Schemas Information

No object selected

tercera clase proyecto Projeto personal

```
-- De la tabla de partes obtén el nombre y la categoría de cada parte a la cual pertenece--  

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 -- 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 "Tematica" 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 count(name) as "cantidad de sets",set_num from inventories where count(name) > 2 group by set_num;
```

Result Grid Filter Rows Export Wrap Cell Content: 15

Nombre del set	Tematica
Horizon Express	RC Train
NHL 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	Clikits
Birthday Pack Daisy	Clikits
Birthday Pack Star	Clikits
High Speed Train Car	World City

Result 4 x

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

Object Info Session Query Completed

# Luis F Castro

LuisCastro.parts Aggregations LuisCastro.inventory\_sets Aggregations

**LuisCastro.inventory\_sets**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proj 2 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 remote join field.
5 * as: The name for the results.
6 * pipeline: The pipeline to run on the joined collection.
7 * let: Optional variables to use in the pipeline if
8 *      $let is used.
9 */
10 from: 'sets',
11 localField: 'set_num',
12 foreignField: 'set_num',
13 as: 'sets'
14

```

**\$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 collection.
7 * let: Optional variables to use in the pipeline if
8 *      $let is used.
9 */
10 from: 'themes',
11 localField: 'sets.theme_id',
12 foreignField: 'id',
13 as: 'theme'
14

```

DOCUMENTS 2.8k TOTAL SIZE 200.6KB AVG. SIZE 74B INDEXES 1 TOTAL SIZE 52.0KB AVG. SIZE 52.0KB

LuisCastro.inventory\_sets

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proj 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   object: {$arrayElemAt:[ "$sets", 0 ]},
7   objitem: {$arrayElemAt:[ "$theme", 0 ]},
8

```

**\$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   {
7     nombre_set: "obj.item.name",
8     año: "obj.item.year",
9     tematica: "obj.item.name"
9

```

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

```

1 /**

```

DOCUMENTS 2.8k TOTAL SIZE 200.6KB AVG. SIZE 74B INDEXES 1 TOTAL SIZE 52.0KB AVG. SIZE 52.0KB

## Luis F Castro

LuisCastro.inventory\_sets

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proj 2 SAVE SAMPLE MODE AUTO PREVIEW

**SaddFields**

Output after **SaddFields** stage (Sample of 20 documents)

```
1 /**
2 * @param field_name: The new field name.
3 * @param expression: The field expression.
4 */
5 +
6 nombre_set: "$objSet.name",
7 año: "$objSet.year",
8 tematica: "$objItem.name"
9 }
```

**project**

Output after **project** stage (Sample of 20 documents)

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

ADD STAGE

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

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 "categoria" from parts as part
6 join part_categories as pc on part.part_cat_id = pc.id order by categoria;
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 insert(part_num,quantity) as "cantidad de partes" sets.name as "nombre del set" themes.name as "Temática"
```

Result Grid Filter Rows: Export: Wrap Cell Content:

total de piezas	is_trans
107	f
28	t

Result 5

LuisCastro.colors

The screenshot shows the MongoDB Aggregations interface. At the top, there are tabs for 'Documents', 'Aggregations' (which is selected), 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. Below the tabs, it says '135 Documents in the Collection'. On the right, it shows 'TOTAL SIZE 11.3KB AVG. SIZE 85B' for documents and 'TOTAL SIZE 4.0KB AVG. SIZE 4.0KB' for indexes. There are also 'SAMPLE MODE' and 'AUTO PREVIEW' buttons.

In the main area, there's a preview of three documents:

- `_id:ObjectId("5fc...")`
- `_id:ObjectId("5fc...")`
- `_id:ObjectId("5fc...")`

Below the preview, a message says "Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)".

Under the '\$group' stage, the pipeline looks like this:

```

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
  
```

The output after the '\$group' stage shows two documents:

- `_id:"t"` `verd:28`
- `_id:"f"` `verd:107`

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

The screenshot shows a MySQL query editor with the following SQL code:

```

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 inventories 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;
  
```

The results grid shows the following data:

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

# Luis F Castro

**LuisCastro.inventory\_sets**

Documents	Aggregations	Schema	Explain Plan	Indexes	Validation	DOCUMENTS 2.8k	TOTAL SIZE 206.8KB	AVG. SIZE 74B	INDEXES 1	TOTAL SIZE 52.0KB	AVG. SIZE 52.0KB
<p><b>\$match</b> <input checked="" type="checkbox"/> Output after \$match stage (Sample of 20 documents)</p> <pre>1 /** 2  * query: The query in MQL. 3  */ 4 [ 5   { 6     quantity:{\$gt:2} 7   } 8 ]</pre>											
<p><b>\$lookup</b> <input checked="" type="checkbox"/> Output after \$lookup stage (Sample of 20 documents)</p> <pre>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 collection. 7  * let: Optional variables to use in the pipeline file. 8 */ 9 [ 10   { 11     from: 'sets', 12     localField: 'set_num', 13     foreignField: 'set_num', 14     as: 'sets' 15   } 16 ]</pre>											
<p><b>\$sort</b> <input checked="" type="checkbox"/> Output after \$sort stage (Sample of 20 documents)</p> <pre>1 /** 2  * Provide any number of field/order pairs. 3  */ 4 [ 5   { 6     quantity: -1 7   } 8 ]</pre>											

**LuisCastro.inventory\_sets**

Documents	Aggregations	Schema	Explain Plan	Indexes	Validation	DOCUMENTS 2.8k	TOTAL SIZE 206.8KB	AVG. SIZE 74B	INDEXES 1	TOTAL SIZE 52.0KB	AVG. SIZE 52.0KB
<p><b>\$lookup</b> <input checked="" type="checkbox"/> Output after \$lookup stage (Sample of 20 documents)</p> <pre>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 collection. 7  * let: Optional variables to use in the pipeline file. 8 */ 9 [ 10   { 11     from: 'themes', 12     localField: 'sets.theme_id', 13     foreignField: 'id', 14     as: 'tematica' 15   } 16 ]</pre>											
<p><b>\$sort</b> <input checked="" type="checkbox"/> Output after \$sort stage (Sample of 20 documents)</p> <pre>1 /** 2  * Provide any number of field/order pairs. 3  */ 4 [ 5   { 6     quantity: -1 7   } 8 ]</pre>											

## Luis F Castro

**LuisCastro.inventory\_sets**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proy 4 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   {
7     objects:[{$arrayElemAt:[ "$sets", 0 ]}],
8     objitem:[{$arrayElemAt:[ "$tematica ", 0 ]}]
9   }
10 ]

```

**\$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   {
7     NombreDelSet: "$objsets.name",
8     Tematica:"$objitem.name"
9   }
10 ]

```

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

**LuisCastro.inventory\_sets**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proy 4 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   {
7     NombreDelSet: "$objsets.name",
8     Tematica:"$objitem.name"
9   }
10 ]

```

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

```

1 /**
2 * specifications: The fields to
3 * include or exclude.
4 */
5 [
6   {
7     quantity:1,
8     NombreDelSet:1,
9     Tematica:1
10 }

```

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

5.-Del inventario de partes, obtenga el color de la parte, el rango rgb, la cantidad en stock y si se vende por separado.

Luis F Castro

The screenshot shows the MySQL Workbench interface with the following details:

- Top Bar:** Bedu7 x, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas Panel:** Shows the current schema is "tercer clase" and "Proyecto personal". It lists several databases and projects under "SCHEMAS".
- Query Editor:** Displays a SQL query and its results. The query joins multiple tables to find parts with specific quantities and colors. The results are as follows:

quantity	is_spare	name	rgb
1	f	Dark Bush Gray	6C7A8B
1	f	Light Gray	A9A9A9
1	f	Orange	FEB818
1	f	Black	05131D
1	f	Light Flesh	F6D7B3
1	f	Trans-Clear	FCFCFC
1	f	Bright Pink	E4A0C8
1	f	Green	237841
1	f	White	FFFFFF
2	f	White	FFFFFF
1	f	White	FFFFFF

- Results Grid:** Shows the data from the query in a tabular format with columns: quantity, is\_spare, name, and rgb.
- Output Panel:** Displays the execution log with entries for each query run.
- Bottom Navigation:** Object Info, Session, Overview Completed.

LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation

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 alias for the joined documents.
6 * pipeline: The pipeline to run on the joined collection.
7 * let: optional variables to use in the pipeline if needed.
8 */
9 *{
10   from: "colors",
11   localField: "color_id",
12   foreignField: "id",
13   as: "col"
14 }
```

**\$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   objrbg: {$arrayElemAt:[ "$col", 0 ]}
7 }
```

DOCUMENTS 580.3k TOTAL SIZE 54.7MB AVERAGE SIZE 99B INDEXES 1 TOTAL SIZE 5.4MB AVERAGE SIZE 5.4MB

LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE D

**\$addFields** (Sample of 20 documents)

```

1 /**
2 * newField: The new field name.
3 * expression: The new field expression.
4 */
5 + {
6   RGB: "$objrgb.rgb",
7   color: "$objrgb.name"
8 }

```

**\_id: Object**  
**inventory\_id: 1**  
**part\_num: "A8379c01"**  
**color\_id: 72**  
**quantity: 1**  
**is\_spare: true**  
**col: Array**  
**RGB: "000080"**  
**color: "Dark Bluish Gray"**

**\_id: Object**  
**inventory\_id: 1**  
**part\_num: "A8379c01"**  
**color\_id: 7**  
**quantity: 1**  
**is\_spare: true**  
**col: Array**  
**RGB: "000080"**  
**color: "Dark Bluish Gray"**

**\$project** (Sample of 20 documents)

```

1 /**
2 * specifications: The fields to include or exclude.
3 */
4 +
5 + {
6   _id: 1,
7   color: 1,
8   is_spare: 1,
9   _id0: 1,
10  quantity: 1
11 }

```

**quantity: 1**  
**is\_spare: true**  
**color: "Dark Bluish Gray"**

**quantity: 1**  
**is\_spare: true**  
**color: "Light Gray"**

**quantity: 1**  
**is\_spare: true**  
**color: "000080"**

ADD STAGE

6.-Obten las cantidades de sets disponibles, primero ordenando por año del más antiguo al más moderno y después por cantidad.

Limit to 1000 rows

```

17 join themes on sets.theme_id=themes.id
18 where ispart.quantity>2;
19 /*Del inventario de partes, obtenga el color de la parte, el rango rgb, la cantidad en stock y si se vende por separado.*/
20 • select quantity,is_spare,name,rgb from inventory_parts ip
21 join colors col on ip.color_id=col.id;
22 /*Obten las cantidades de sets disponibles, primero ordenando por año del más antiguo al más moderno y después por cantidad.*/
23 • select name,year,quantity from sets
24 join inventory_sets iset on sets.set_num=iset.set_num order by year,quantity desc;
25

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

name	year	quantity
Weetabix Promotional House 1	1976	1
Weetabix Promotional House 2	1976	1
Weetabix Promotional Windmill	1976	1
NHL Action Set with Stickers	2003	2
High Speed Train Car	2004	2
TIE Fighter Collection	2004	1
Birthday Pack Heart	2004	1
Birthday Pack Daisy	2004	1
Birthday Pack Star	2004	1
Horizon Express	2013	1

## Luis F Castro

**LuisCastro.inventory\_sets**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: proy 6 - Modified | SAVE | SAMPLE MODE | AUTO PREVIEW

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

```

1 var {
2   from: 'sets',
3   localField: 'set_num',
4   foreignField: 'set_num',
5   as: 'iset'
6 }

```

```

_id: ObjectId("5fcae538098f797d78afa685")
inventory_id: 35
set_num: "75911-1"
quantity: 1
iset: Array
  0: Object
    _id: ObjectId("5fcae53cb098f797d78b03807")
    set_num: "75911-1"
    name: "McLaren Mercedes Pit Stop"

```

```

_id: ObjectId("5fcae538098f797d78afa686")
inventory_id: 35
set_num: "75912-1"
quantity: 1
iset: Array
  0: Object
    _id: ObjectId("5fcae53cb098f797d78b03807")
    set_num: "75911-1"
    name: "McLaren Mercedes Pit Stop"

```

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

```

1 var {
2   objset: {
3     $arrayElemAt: [
4       '$iset',
5       0
6     ]
7   }
8 }

```

```

_id: ObjectId("5fcae538098f797d78afa685")
inventory_id: 35
set_num: "75911-1"
quantity: 1
iset: Array
objset: Object

```

```

_id: ObjectId("5fcae538098f797d78afa686")
inventory_id: 35
set_num: "75912-1"
quantity: 1
iset: Array
objset: Object

```

**LuisCastro.inventory\_sets**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: proy 6 - Modified | SAVE | SAMPLE MODE | AUTO PREVIEW

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

```

1 var {
2   Nombre: '$objset.name',
3   año: '$objset.year'
4 }

```

```

_id: ObjectId("5fcae538098f797d78afa685")
inventory_id: 35
set_num: "75911-1"
quantity: 1
iset: Array
objset: Object
Nombre: "McLaren Mercedes Pit Stop"
año: 2015

```

```

_id: ObjectId("5fcae538098f797d78afa686")
inventory_id: 35
set_num: "75912-1"
quantity: 1
iset: Array
objset: Object
Nombre: "Porsche 911 GT Finish Line"
año: 2015

```

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

```

1 var {
2   quantity: 1,
3   nombre: 1,
4   año: 1,
5   Nombre: 1
6 }

```

```

_id: ObjectId("5fcae538098f797d78afa685")
quantity: 1
Nombre: "McLaren Mercedes Pit Stop"
año: 2015

```

```

_id: ObjectId("5fcae538098f797d78afa686")
quantity: 1
Nombre: "Porsche 911 GT Finish Line"
año: 2015

```

## Luis F Castro

**LuisCastro.inventory\_sets**

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 2.8k TOTAL SIZE 206.8KB AVG. SIZE 74B INDEXES 1 TOTAL SIZE 52.0KB AVG. SIZE 52.0KB

**Aggregations Stage:**

```

1 ▶ {
2   quantity: 1,
3   nombre:1,
4   año:1,
5   Nombre:1
6 }

```

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

<code>_id: ObjectId("5fc...")</code> quantity:1 Nombre: "McLaren Mercedes Pit Stop" año: 2015	<code>_id: ObjectId("5fc...")</code> quantity:1 Nombre: "Porsche 911 GT Finish Line" año: 2015	<code>_id: ObjectId("5fc...")</code> quantity:1 Nombre: "T... año: 2014
--------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------

**Sort Stage:**

```

1 ▶ /**
2  * Provide any number of field/order pairs.
3  */
4 ▶ {
5   año:1,quantity:-1
6 }

```

**Error in \$cursor stage :: caused by :: operation exceeded time limit**

**No Preview Documents**

**ADD STAGE**

**SQL Query:**

```

16 join sets on inv.set_num=set.set_num
17 join themes on sets.theme_id=themes.id
18 where ipart.quantity>2;
19 /*Del inventario de partes, obtenga el color de la parte, el rango rgb, la cantidad en stock y si se vende por separado.*/
20 • select quantity,is_spare,name,rgb from inventory_parts ip
21 join colors col on ip.color_id=col.id;
22 /*Obten las cantidades de sets disponibles, primero ordenando por año del más antiguo al más moderno y despues por cantidad.*/
23 • select name,year,quantity from sets
24 join inventory_sets iset on sets.set_num=iset.set_num order by year,quantity desc;
25

```

**Result Grid:**

name	year	quantity
Weetabix Promotional House 1	1976	1
Weetabix Promotional House 2	1976	1
Weetabix Promotional Windmill	1976	1
NHL Action Set with Stickers	2003	2
High Speed Train Car	2004	2
TIE Fighter Collection	2004	1
Birthday Pack Heart	2004	1
Birthday Pack Daisy	2004	1
Birthday Pack Star	2004	1
Horizon Express	2013	1

### 7.-Consulta cuantas partes hay por categoría y ordénalas de mayor a menor.

**SQL Query:**

```

19 /*Del inventario de partes, obtenga el color de la parte, el rango rgb, la cantidad en stock y si se vende por separado.*/
20 • select quantity,is_spare,name,rgb from inventory_parts ip
21 join colors col on ip.color_id=col.id;
22 /*Obten las cantidades de sets disponibles, primero ordenando por año del más antiguo al más moderno y despues por cantidad.*/
23 • select name,year,quantity from sets
24 join inventory_sets iset on sets.set_num=iset.set_num order by year,quantity desc;
25 /*De las categorías de partes realiza un conteo de cuantas partes pertenecen a cada categoría*/
26 • select pc.name,count(part.name) from parts as part
27 join part_categories as pc on part.part_cat_id = pc.id group by pc.name order by count(part.name) desc;

```

**Result Grid:**

name	count(part.name)
Non-LEGO	88
Minifig Accessories	49
Minifigs	36
Baseplates	17
Duplo, Quattro an...	10
Other	9
Rock	9
Plants and Animals	7
Windscreens and...	4
Containers	3

## Luis F Castro

### LuisCastro.parts

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 26.0k TOTAL SIZE 3.2MB AVG. SIZE 129B INDEXES 1 TOTAL SIZE 268.0KB AVG. SIZE 268.0KB

**\$lookup**

```
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 collection.
7  * let: Optional variables to use in the pipeline if
8  */
9 +{
10  from: 'part_categories',
11  localField: 'part_cat_id',
12  foreignField: 'id',
13  as: 'pcat'
14 }
```

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

```
_id:ObjectId("5fciae5750908f797d78afb1dc")
part_num:"0687b1"
name:"Set 0687 Activity Booklet 1"
part_cat_id:17
pcat:Array
```

```
_id:ObjectId("5fciae5750908f797d78afb1dd")
part_num:"0901"
name:"Baseplate 16 x 30 with Set 080 Yellow House
Print"
part_cat_id:1
pcat:Array
```

```
_id:Object
part_num:''
name:"Base
Print
part_cat_i
pcat:Array
pcat:Obj
```

**\$addFields**

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

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

```
_id:ObjectId("5fciae5750908f797d78afb1dc")
part_num:"0687b1"
name:"Set 0687 Activity Booklet 1"
part_cat_id:17
pcat:Array
objcat:Object
```

```
_id:ObjectId("5fciae5750908f797d78afb1dd")
part_num:"0901"
name:"Baseplate 16 x 30 with Set 080 Yellow House
Print"
part_cat_id:1
pcat:Array
objcat:Object
```

```
_id:Object
part_num:''
name:"Base
Print
part_cat_i
pcat:Array
objcat:Obj
```

### LuisCastro.parts

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 26.0k TOTAL SIZE 3.2MB AVG. SIZE 129B INDEXES 1 TOTAL SIZE 268.0KB AVG. SIZE 268.0KB

**\$addFields**

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

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

```
_id:ObjectId("5fciae5750908f797d78afb1dc")
part_num:"0687b1"
name:"Set 0687 Activity Booklet 1"
part_cat_id:17
pcat:Array
objcat:Object
categoria:"Non-LEGO"
```

```
_id:ObjectId("5fciae5750908f797d78afb1dd")
part_num:"0901"
name:"Baseplate 16 x 30 with Set 080 Yellow House
Print"
part_cat_id:1
pcat:Array
objcat:Object
categoria:"Baseplates"
```

```
_id:Object
part_num:''
name:"Base
Print
part_cat_i
pcat:Array
objcat:Obj
categoria:
```

**\$group**

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

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

```
_id:"Technic Beams"
Numeropartes:32
```

```
_id:"Mechanical"
Numeropartes:28
```

```
_id:"Bars,
Numeropart
```

## Luis F Castro

LuisCastro.parts

The screenshot shows the MongoDB Aggregation interface with two stages displayed:

- \$group stage:** Shows a sample of 20 documents grouped by category. One document example is shown: `_id: "Technic Beams"` with `Numeropartes: 32`.
- \$sort stage:** Shows a sample of 20 documents sorted by Numeropartes. One document example is shown: `_id: "Minifigs"` with `Numeropartes: 8556`.

8.-Consulta de qué set son todas las piezas que se venden por separado y ordénalas de mayor a menor.

The screenshot shows the MySQL Workbench SQL editor with the following query:

```

25 /*De las categorías de partes realiza un conteo de cuantas partes pertenecen a cada categoría*/
26 • select pc.name,count(part.name) from parts as part
27 join part_categories as pc on part.part_cat_id = pc.id group by pc.name order by count(part.name) desc;
28 /*Consulta de qué set son todas las piezas que se venden por separado y ordénalas de mayor a menor*/
29 • select /*,count(part_num)*/ from inventory_parts ip
30 join inventories inv on ip.inventory_id=inv.id
31 join sets on inv.set_num=sets.set_num
32 where is_spare='t'
33 /*group by name*/;
```

The result grid shows one row of data:

inventory_id	part_num	color_id	quantity	is_spare	id	version	set_num	set_num	name	year	theme_id	num_parts
49	6141	15	1	t	49	1	10071-1	10071-1	Mr. Bunny	2003	229	25

The screenshot shows the MySQL Workbench SQL editor with the following query:

```

25 /*De las categorías de partes realiza un conteo de cuantas partes pertenecen a cada categoría*/
26 • select pc.name,count(part.name) from parts as part
27 join part_categories as pc on part.part_cat_id = pc.id group by pc.name order by count(part.name) desc;
28 /*Consulta de qué set son todas las piezas que se venden por separado y ordénalas de mayor a menor*/
29 • select count(part_num) from inventory_parts ip
30 join inventories inv on ip.inventory_id=inv.id
31 join sets on inv.set_num=sets.set_num
32 where is_spare='t'
33 group by name;
```

The result grid shows one row of data:

count(part_num)
1

## Luis F Castro

**LuisCastro.inventory\_parts**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proy 8 SAVE SAMPLE MODE AUTO PREVIEW

**\$match** Output after \$match stage (Sample of 20 documents)

```
1: {
  2:   "is_spare": "t"
  3: }
```

`_id: ObjectId("5fcdbaille957ac284035763b")`
`inventory_id: 3`
`part_num: "33291"`
`color_id: 191`
`quantity: "1"`
`is_spare: "t"`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`

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

```
1: {
  2:   "from": "inventories",
  3:   "localField": "inventory_id",
  4:   "foreignField": "id",
  5:   "as": "inv"
  6: }
```

`_id: ObjectId("5fcdbaille957ac284035763b")`
`inventory_id: 3`
`part_num: "33291"`
`color_id: 191`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`

**LuisCastro.inventory\_parts**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proy 8 SAVE SAMPLE MODE AUTO PREVIEW

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

```
1: {
  2:   "from": "sets",
  3:   "localField": "inv.set_num",
  4:   "foreignField": "set_num",
  5:   "as": "set"
  6: }
```

`_id: ObjectId("5fcdbaille957ac284035763b")`
`inventory_id: 3`
`part_num: "33291"`
`color_id: 191`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`
`set: Array`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`
`set: Array`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`
`set: Array`

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

```
1: {
  2:   "obj": {
  3:     "$arrayElemAt": [
  4:       "$set",
  5:       0
  6:     ]
  7:   }
  8: }
```

`_id: ObjectId("5fcdbaille957ac284035763b")`
`inventory_id: 3`
`part_num: "33291"`
`color_id: 191`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`
`set: Array`
`obj: Object`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`
`set: Array`
`obj: Object`

`_id: ObjectId("5fcdbaille957ac2840357643")`
`inventory_id: 3`
`part_num: "6141"`
`color_id: 27`
`quantity: "1"`
`is_spare: "t"`
`inv: Array`
`set: Array`
`obj: Object`

## Luis F Castro

LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 580.3k TOTAL SIZE 58.6MB AVG. SIZE 106B INDEXES 1 TOTAL SIZE 5.0MB AVG. SIZE 5.0MB

`$addFields`

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

```

1 ↵ [
2   ↵   Set: '$obj.name'
3   ↵ ]

```

```

inventory_xu: 3
part_num: "33291"
color_id: 191
quantity: "1"
is_spare: "t"
▶ inv: Array
▶ set: Array
▶ obj: Object
Set: "Emma's Splash Pool"

```

```

_id: ObjectId("5fcdbba11e957ac2840357643")
inventory_id: 3
part_num: "6141"
color_id: 27
quantity: "1"
is_spare: "t"
▶ inv: Array
▶ set: Array
▶ obj: Object

```

```

_id: ObjectId("5fcdbba11e957ac2840357643")
inventory_id: 3
part_num: "6141"
color_id: 27
quantity: "1"
is_spare: "t"
▶ inv: Array
▶ set: Array
▶ obj: Object

```

`$limit`

Output after `$limit` stage (Sample of 20 documents)

```

1 150

```

```

_id: ObjectId("5fcdbba11e957ac284035763b")
inventory_id: 3
part_num: "33291"
color_id: 191
quantity: "1"
is_spare: "t"
▶ inv: Array
▶ set: Array
▶ obj: Object

```

```

_id: ObjectId("5fcdbba11e957ac2840357643")
inventory_id: 3
part_num: "6141"
color_id: 27
quantity: "1"
is_spare: "t"
▶ inv: Array
▶ set: Array
▶ obj: Object

```

```

_id: ObjectId("5fcdbba11e957ac2840357643")
inventory_id: 3
part_num: "6141"
color_id: 27
quantity: "1"
is_spare: "t"
▶ inv: Array
▶ set: Array
▶ obj: Object

```

LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 580.3k TOTAL SIZE 58.6MB AVG. SIZE 106B INDEXES 1 TOTAL SIZE 5.0MB AVG. SIZE 5.0MB

`$group`

Output after `$group` stage (Sample of 20 documents)

```

1 ↵ [
2   ↵   _id: '$$Set',
3   ↵   numero_Sets: {
4     ↵     $sum: 1
5   ↵ }
6   ↵ ]

```

```

_id: "X-wing Fighter - Mini (Kabaya Box)"
numero_Sets: 1

```

```

_id: "Taj Mahal"
numero_Sets: 10

```

```

_id: "Adver
numero_Set

```

`$sort`

Output after `$sort` stage (Sample of 20 documents)

```

1 ↵ [
2   ↵   numero_Sets: -1
3   ↵ ]

```

```

_id: "Slave I"
numero_Sets: 33

```

```

_id: "Creative Ambush"
numero_Sets: 18

```

```

_id: "Taj Mahal"
numero_Sets: 10

```

9.- Consulta de qué set son todas las piezas que no se venden por separado y ordénalas de mayor a menor.

## Luis F Castro

```

31   join sets on inv.set_num=sets.set_num
32   where is_spare="t"
33   group by name;
34   /*Consulta de qué set son todas las piezas que no se venden por separado y ordénalas de mayor a menor*/
35 •   select count(part_num) from inventory_parts ip
36   join inventories inv on ip.inventory_id=inv.id
37   join sets on inv.set_num=sets.set_num
38   where is_spare="f"
39   group by name;

```

The screenshot shows a MongoDB query editor interface. At the top, there's a toolbar with various icons. Below it is a code editor window displaying the provided aggregation pipeline. At the bottom, there's a result grid showing a single row with the value 'count(part\_num)' and a count of '17'.

### LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation SAMPLE MODE AUTO PREVIEW

**\$match** Output after \$match stage (Sample of 20 documents)

```

1 {
2   is_spare: "f"
3 }

```

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

```

1 {
2   from: 'inventories',
3   localField: 'inventory_id',
4   foreignField: 'id',
5   as: 'inv'
6 }

```

This section shows the results of the \$match and \$lookup stages of the aggregation pipeline. It displays sample documents and their corresponding joined inventories.

### LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation SAMPLE MODE AUTO PREVIEW

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

```

1 {
2   from: 'sets',
3   localField: 'inv.set_num',
4   foreignField: 'set_num',
5   as: 'set'
6 }

```

**\$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   obj: {$arrayElemAt:[ "$set", 0 ]}
7 }

```

This section shows the results of the \$lookup and \$addFields stages. It displays sample documents and their joined sets, along with newly added fields.

## Luis F Castro

**LuisCastro.inventory\_parts**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proj 9 SAVE SAMPLE MODE AUTO PREVIEW

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

```

1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4 */
5 [
6   Set: "$obj.name"
7 ]
  
```

part\_num: 48395
color\_id: 72
quantity: "1"
is\_spare: "f"
inv: Array
set: Array
obj: Object
Set: "McDonald's Sports Set Number 6 - Orange Vest
Snowboarder"

\_id: ObjectId("5fcdbaille957ac284035762a")
inventory\_id: 1
part\_num: "48395"
color\_id: 7
quantity: "1"
is\_spare: "f"
inv: Array
set: Array
obj: Object

\_id: Object
inventory\_
part\_num: ''
color\_id: ''
quantity: ''
is\_spare: ''
inv: Array
set: Array
obj: Object

**Output after \$limit stage (Sample of 20 documents)**

```

1 /**
2  * Provide the number of documents to limit.
3  */
4 150
  
```

\_id: ObjectId("5fcdbaille957ac2840357629")
inventory\_id: 1
part\_num: "48379c61"
color\_id: 72
quantity: "1"
is\_spare: "f"
inv: Array
set: Array
obj: Object

\_id: ObjectId("5fcdbaille957ac284035762a")
inventory\_id: 1
part\_num: "48395"
color\_id: 7
quantity: "1"
is\_spare: "f"
inv: Array
set: Array
obj: Object

\_id: Object
inventory\_
part\_num: ''
color\_id: ''
quantity: ''
is\_spare: ''
inv: Array
set: Array
obj: Object

**LuisCastro.inventory\_parts**

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION proj 9 SAVE SAMPLE MODE AUTO PREVIEW

**Output after \$group stage (Sample of 11 documents)**

```

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

\_id: "Emma's Splash Pool"
numero\_Sets: 26

\_id: "McDonald's Sports Set Number 6 - Orange Vest
Snowboarder"
numero\_Sets: 4

\_id: "1 x f
numero\_Set

**Output after \$sort stage (Sample of 11 documents)**

```

1 /**
2  * Provide any number of field/order pairs.
3 */
4 [
5   numero_Sets: -1
6 ]
  
```

\_id: "Coast Guard HQ"
numero\_Sets: 43

\_id: "First Order Star Destroyer"
numero\_Sets: 27

\_id: "Emma"
numero\_Set

10.- Crea una consulta empezando por la tabla partes de inventarios que incluya, el numero de parte, la cantidad, si se vende suelta, el color, el rgb, si es transparente, el set al que pertenece, el año del set y la temática.

## Luis F Castro

38       where is\_spare="f"  
 39       group by name  
 40       /\*Crea una consulta empezando por la tabla partes de inventarios que incluya, el numero de parte, la cantidad,  
 41       si se vende suelta, el color, el rgb, si es transparente, el set al que pertenece, el año del set y la temática.\*/  
 42       select part\_num,quantity,is\_spare,version,sets.name,year,themes.name,col.name,rgb,is\_trans from inventory\_parts ip  
 43       join inventories inv on ip.inventory\_id=inv.id  
 44       join sets on inv.set\_num=sets.set\_num  
 45       join themes on sets.theme\_id= themes.id  
 46       join colors col on ip.color\_id=col.id;

part_num	quantity	is_spare	version	name	year	name	name	rgb	is_trans
3665	2	f	1	Mr. Bunny	2003	Easter	Blue	0055BF	f
3040b	2	f	1	Mr. Bunny	2003	Easter	Blue	0055BF	f
3020	1	f	1	Mr. Bunny	2003	Easter	Blue	0055BF	f
3004	1	f	1	Mr. Bunny	2003	Easter	Blue	0055BF	f
3710	1	f	1	Mr. Bunny	2003	Easter	Red	C91A09	f
3039	1	f	1	Mr. Bunny	2003	Easter	Red	C91A09	f
3023	1	f	1	Mr. Bunny	2003	Easter	Red	C91A09	f
3022	1	f	1	Mr. Bunny	2003	Easter	Red	C91A09	f
4286	3	f	1	Mr. Bunny	2003	Easter	Yellow	F2CD37	f
3710	2	f	1	Mr. Bunny	2003	Easter	Yellow	F2CD37	f

LuisCastro.inventory\_parts

Documents Aggregations Schema Explain Plan Indexes Validation

SAVE SAMPLE MODE AUTO PREVIEW

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 collection.
7  * let: Optional variables to use in the pipeline f
8 */
9 *
10 from: 'colors',
11 localField: 'color_id',
12 foreignField: 'id',
13 as: 'col'
14 }

```

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 collection.
7  * let: Optional variables to use in the pipeline f
8 */
9 *
10 from: 'inventories',
11 localField: 'inventory_id',
12 foreignField: 'id',
13 as: 'inv'
14 }

```

## Luis F Castro

**LuisCastro.inventory\_parts**

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 580.3k TOTAL SIZE 58.6MB AVG. SIZE 106B INDEXES 1 TOTAL SIZE 5.0MB AVG. SIZE 5.0MB

**\$lookup** (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 collection.
7  * let: Optional variables to use in the pipeline f
8  */
9 {
10  from: 'sets',
11  localField: 'inv.set_num',
12  foreignField: 'set_num',
13  as: 'set'
14 }

```

**\$lookup** (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 collection.
7  * let: Optional variables to use in the pipeline f
8  */
9 {
10  from: 'themes',
11  localField: 'set.theme_id',
12  foreignField: 'id',
13  as: 'tem'
14 }

```

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

**LuisCastro.inventory\_parts**

Documents Aggregations Schema Explain Plan Indexes Validation DOCUMENTS 580.3k TOTAL SIZE 58.6MB AVG. SIZE 106B INDEXES 1 TOTAL SIZE 5.0MB AVG. SIZE 5.0MB

**\$addFields** (Sample of 20 documents)

```

1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6  objcol: {$arrayElemAt: ["$col", 0]},
7  objset: {$arrayElemAt: ["$set", 0]},
8  objitem: {$arrayElemAt: ["$tem", 0]}
9 }

```

**\$addFields** (Sample of 20 documents)

```

1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6  color: "$objcol.name",
7  rgb: "$objcol.rgb",
8  transp: "$objcol.is_trans",
9  Set: "$objset.name",
10 aho: "$objset.year",
11 tematica: "$objitem.name"
12 }

```

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

## Luis F Castro

LuisCastro.inventory\_parts

The screenshot shows the MongoDB Compass interface with the database 'LuisCastro' and collection 'inventory\_parts'. It displays two stages of a pipeline:

- \$addFields Stage:** Shows a code block for adding fields. The output documents show fields like '\_id', 'inventory\_id', 'part\_num', 'color\_id', 'quantity', 'is\_spare', 'col', 'inv', and 'set'.
- \$project Stage:** Shows a code block for specifying fields to include or exclude. The output documents show fields like 'part\_num', 'color', 'rgb', 'transparente', 'Set', 'año', 'Set1', 'tematica', and 'part\_num' again.

11-15 convierte 5 de las consultas anteriores en vistas.

The screenshot shows the MongoDB Compass interface with the database 'LuisCastro' and collection 'parts'. It displays the creation of five views:

- vista1:** A view that joins parts and part\_categories collections to get part name and category.
- vista2:** A view that counts total pieces per part.
- vista3:** A view that joins inventory\_parts and colors collections to get quantity, is\_spare, name, and color.
- vista4:** A view that counts parts per category.
- vista5:** A view that joins inventory\_parts and inventories collections to get part number, name, and category, filtering for is\_spare = 'f'.

The bottom section shows the results for 'vista 1' (view on LuisCastro.parts), displaying documents with part numbers, names, and categories.

# Luis F Castro

The screenshots demonstrate the use of MongoDB's aggregation framework across three different collections.

**LuisCastro.vista 2 (view on: LuisCastro.colors)**

This view shows the aggregation results for the `LuisCastro.colors` collection. It displays two documents:

```
_id: "t"
verd: 28
```

```
_id: "f"
verd: 187
```

A sample of the aggregated results from this stage is shown below, indicating no preview documents.

**LuisCastro.vista 3 (view on: LuisCastro.inventory\_parts)**

This view shows the aggregation results for the `LuisCastro.inventory_parts` collection. It displays five documents:

```
quantity: "1"
is_spare: "t"
color: "Bright Light Orange"
```

```
quantity: "1"
is_spare: "f"
color: "white"
```

```
quantity: "1"
is_spare: "f"
color: "Light Flesh"
```

```
quantity: "5"
is_spare: "f"
color: "Lime"
```

```
quantity: "1"
is_spare: "f"
color: "white"
```

**LuisCastro.vista 4 (view on: LuisCastro.parts)**

This view shows the aggregation results for the `LuisCastro.parts` collection. It displays five documents:

```
_id: "Bars, Ladders and Fences"
NumeroPartes: 115
```

```
_id: "Baseplates"
NumeroPartes: 250
```

```
_id: "Belville, Scala and Fabuland"
NumeroPartes: 234
```

```
_id: "Blonicle, Hero Factory and Construction"
NumeroPartes: 1116
```

```
_id: "Bricks"
NumeroPartes: 93
```

```
_id: "Bricks Curved"
NumeroPartes: 275
```

## Luis F Castro

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar (Local):** Lists collections: HOSTS, CLUSTER, EDITION, and a search bar for "Filter your data".
- Top Bar:** Shows the database name "LuisCastro" and collection name "inventory\_parts".
- Header:** "LuisCastro.vista 5 (view on: LuisCastro.inventory\_parts)" and "MODIFY SOURCE".
- Header Buttons:** "Read Only", "OPTIONS", "FIND", "RESET", and "REFRESH".
- Document List:** Displays 11 documents. Each document has an "\_id" field and a "numero\_Sets" field.
- Document Examples:**
  - "\_id: "1 x 6 Light Gray Bricks", numero\_Sets:1"
  - "\_id: "Christmas Cat Ornament", numero\_Sets:18"
  - "\_id: "Coast Guard HQ", numero\_Sets:43"
  - "\_id: "Coca-Cola Defenders", numero\_Sets:4"
  - "\_id: "Emma's Splash Pool", numero\_Sets:26"
  - "\_id: "First Order Star Destroyer", numero\_Sets:27"
- Bottom Status:** "Displaying documents 1 - 11 of 11".

## Conclusión

1.- Según las consultas realizadas de las bases de datos podemos afirmar que los inventarios que contienen pertenecen a una tienda, no perteneces a un gran almacén.

2.-El stock se mantiene con números promedio, esto indica que la oferta y la demanda están equilibradas para el tema de los “Sets” salvo por los paquetes que cuyo contenido es misterioso, es decir los sets que tienen piezas pertenecientes a algún tema en específico, pero el contenido fue hecho de piezas al azar, este modelo de negocios impulsa al comprador a seguir comprando los paquetes hasta obtener una pieza en especial.

3.-Algunos “Sets” comparten piezas por pertenecer a la misma temática, una propuesta para ventas puede ser juntar en una promoción los “Sets” de años pasados en conjunto con los “Sets” modernos, esta propuesta permite por ambas partes recibir una ganancia, por el lado del cliente recibe 2 productos con funcionalidad parecida y aprovechar las sinergias de ambos sets mientras que el vendedor comienza a vender “Sets” con poca probabilidad de venta.

4.-Los productos Lego tienen gran versatilidad en sus temáticas y sus propuestas de productos tienen muchas sinergias entre sí por esto mismo es buena idea sugerir un programa de lealtad.

5.-La Base de datos tiene mucha información para brindar pero al tratarse de un inventario es recomendable ampliar la base de datos agregando precios y sacar mucho valor del las bases de datos.

## Bibliografía

<https://www.kaggle.com/rtatman/lego-database>