

Proyecto Bedu (Data)

Título de tu proyecto

Producción y Demoras de Camiones Mineros (CAT 789 y CAT 793)

¿Cuál es la necesidad o problema por resolver? 🚀

Analizar las demoras que generan los camiones fuera de carretera en una operación minera a cielo abierto en la cual se extrae el mineral de Hierro Magnético, así como la producción en toneladas que extrae cada equipo a detalle (por día, turno, equipo, operador, etc).

¿Cuál es la propuesta para resolver esa necesidad o problema? 💡

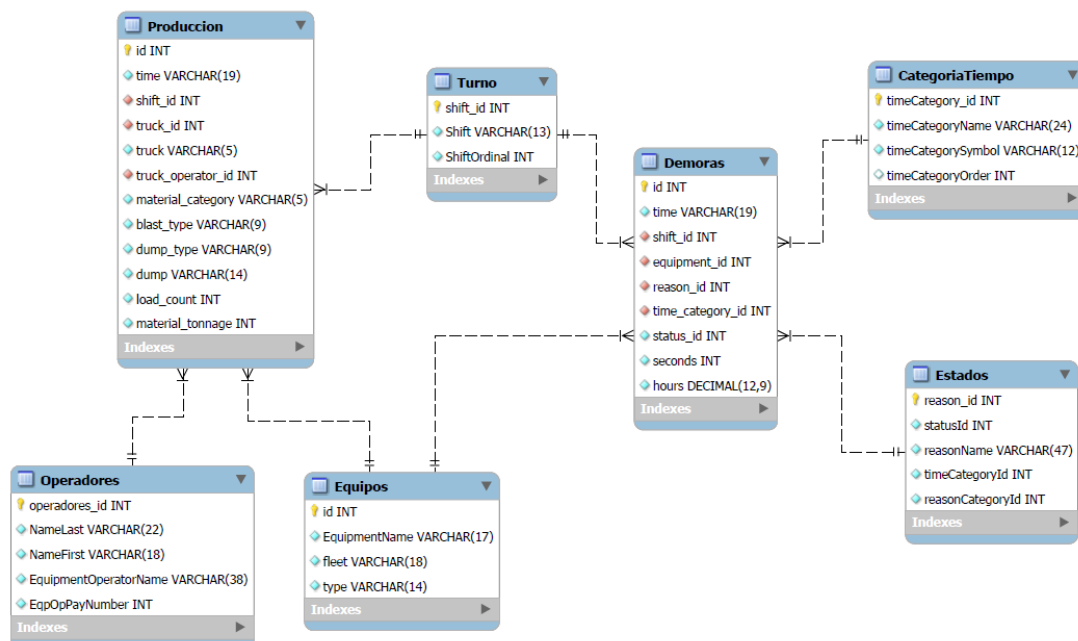
Consultando una base de datos del Sistema de Gestión de Flota utilizado en la operación minera la cual genera información en tiempo real de los camiones mediante el uso de GPS. Para ello pretendo usar MySQL WorkBench ya que todas las tablas tienen relación entre sí.

¿Cómo lo vas a hacer? 📋

Se realizaron 15 preguntas la cual se resuelven mediante consultas tanto en MySQL WorkBench como en MongoDB, la cuales se enlistan a continuación:

1. Dentro de la tabla Operadores, obtén el nombre y apellido de todos los Operadores.
2. Dentro de la tabla Equipos, obtén los equipos de acarreo.
3. Dentro de la tabla Equipos, obtén los camiones modelo CAT 793F
4. ¿Cuál es el nombre del estado el cual tiene símbolo "Down"?
5. Dentro de la tabla Producción, obtener los viajes de mineral que fueron a la trituradora.
6. Dentro de la tabla Operadores, obtener los operadores con apellido Silva.
7. Dentro de la tabla Operadores, obtener los operadores que su nombre termina con "el".
8. Dentro de la tabla demoras, obtén las horas de demora con código 939 por camión.
9. Dentro de la tabla demoras, obtener el top 5 de demoras en total
10. Obtén el máximo, mínimo y promedio de las horas de demora de "Transito"
11. Muestra la información del camión desde la tabla demoras
12. Muestra la información de categoría de tiempo desde la tabla demoras
13. Obtén el nombre del equipo, familia del equipo, nombre de demoras y horas de demora
14. Obtén una vista con el total de horas de alimentos por camión ordenando por el mayor.
15. Obtén una vista con la producción por camión CAT 793F donde incluya nombre de camión, Operador y tipo de material

Diagrama Entidad-Relación



Resultados

SQL Queries

1. Dentro de la tabla Operadores, obtén el nombre y apellido de todos los Operadores.

```
SELECT
    NameFirst, NameLast
FROM
    Operadores;
```

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

```
1 • SELECT
2     NameFirst, NameLast
3 FROM
4     Operadores;
```

The results are displayed in a table with the following data:

NameFirst	NameLast
J Carmen	Rios Gutierrez
J Jesus	Herrera Corderas
Jos Antonio	Fabian Silva
Tom	Coblen Monroy

The bottom section shows the output log with the following entry:

#	Time	Action	Message	Duration / Fetch
1	12:52:52	SELECT	NameFirst, NameLast FROM Oper...	276 row(s) returned 0.078 sec / 0.000 sec

2. Dentro de la tabla Equipos, obtén los equipos de acarreo.

```
SELECT
    *
FROM
    Equipos
WHERE
    type = 'Truck';
```

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

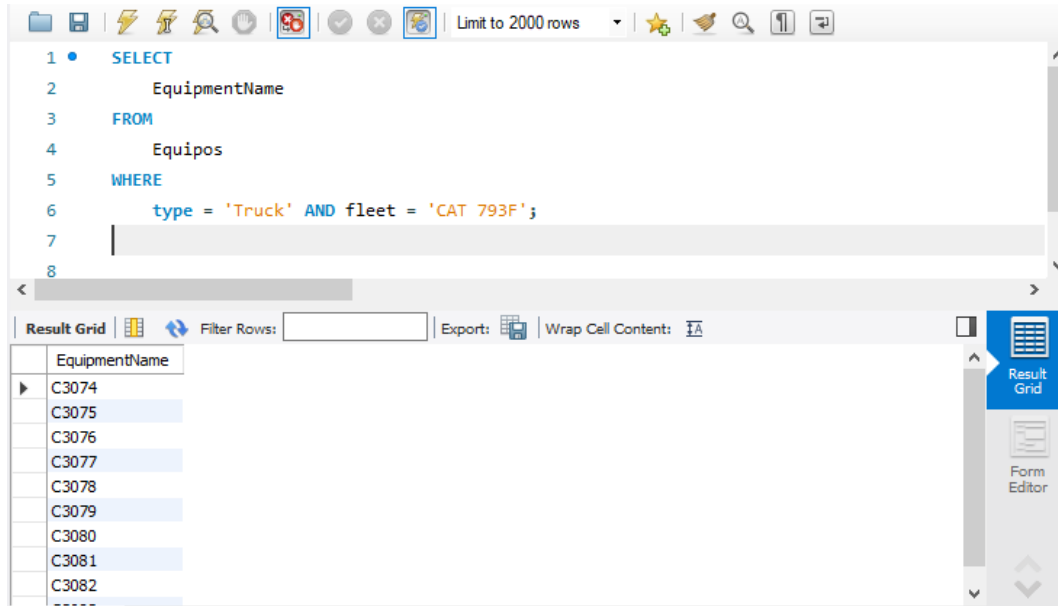
```
1 • SELECT
2     *
3 FROM
4     Equipos
5 WHERE
6     type = 'Truck';
```

The results are displayed in a table with the following data:

id	EquipmentName	fleet	type
447	C3056	CAT 789C	Truck
448	C3057	CAT 789C	Truck
449	C3058	CAT 789C	Truck
450	C3059	CAT 789C	Truck
451	C3060	CAT 789C	Truck
452	C3061	CAT 789C	Truck
453	C3062	CAT 789C	Truck
454	C3063	CAT 789C	Truck
455	C3064	CAT 789C	Truck

3. Dentro de la tabla Equipos, obtener los camiones modelo CAT 793F

```
SELECT
    EquipmentName
FROM
    Equipos
WHERE
    type = 'Truck' AND fleet = 'CAT 793F';
```



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

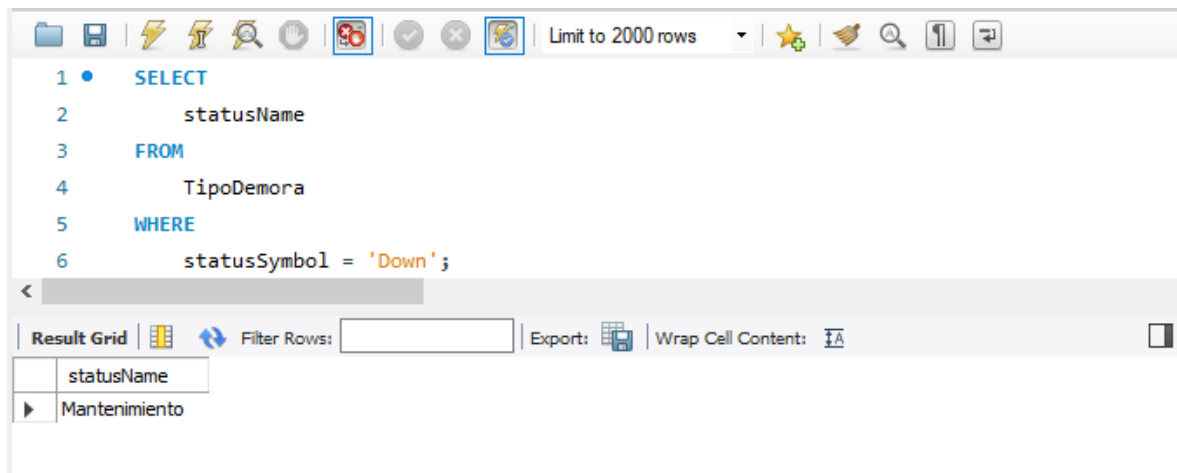
```
1 • SELECT
2     EquipmentName
3 FROM
4     Equipos
5 WHERE
6     type = 'Truck' AND fleet = 'CAT 793F';
7
8
```

Below the query editor is the "Result Grid" showing the results of the query. The grid has one column, "EquipmentName", and lists the following values:

EquipmentName
C3074
C3075
C3076
C3077
C3078
C3079
C3080
C3081
C3082
.....

4. Cual es el nombre del estado el cual tiene símbolo "Down"?

```
SELECT
    statusName
FROM
    TipoDemora
WHERE
    statusSymbol = 'Down';
```



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

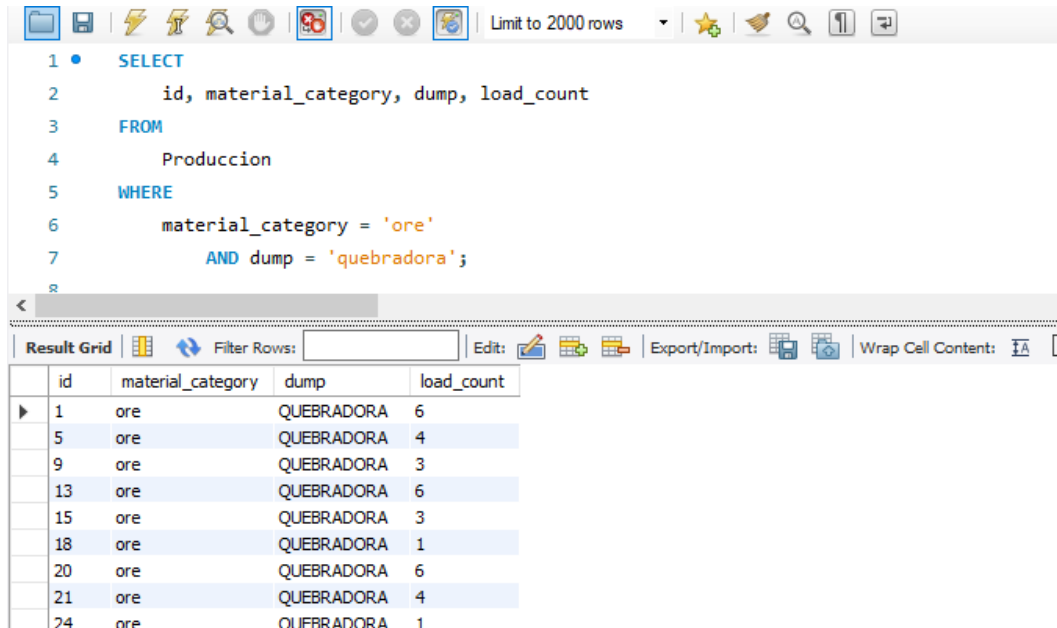
```
1 • SELECT
2     statusName
3 FROM
4     TipoDemora
5 WHERE
6     statusSymbol = 'Down';
```

Below the query editor is the "Result Grid" showing the results of the query. The grid has one column, "statusName", and lists the following value:

statusName
Mantenimiento

5. Dentro de la tabla Producción, obtener los viajes de mineral que fueron a la trituradora.

```
SELECT
    id, material_category, dump, load_count
FROM
    producción
WHERE
    material_category = 'ore'
    AND dump = 'quebradora';
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

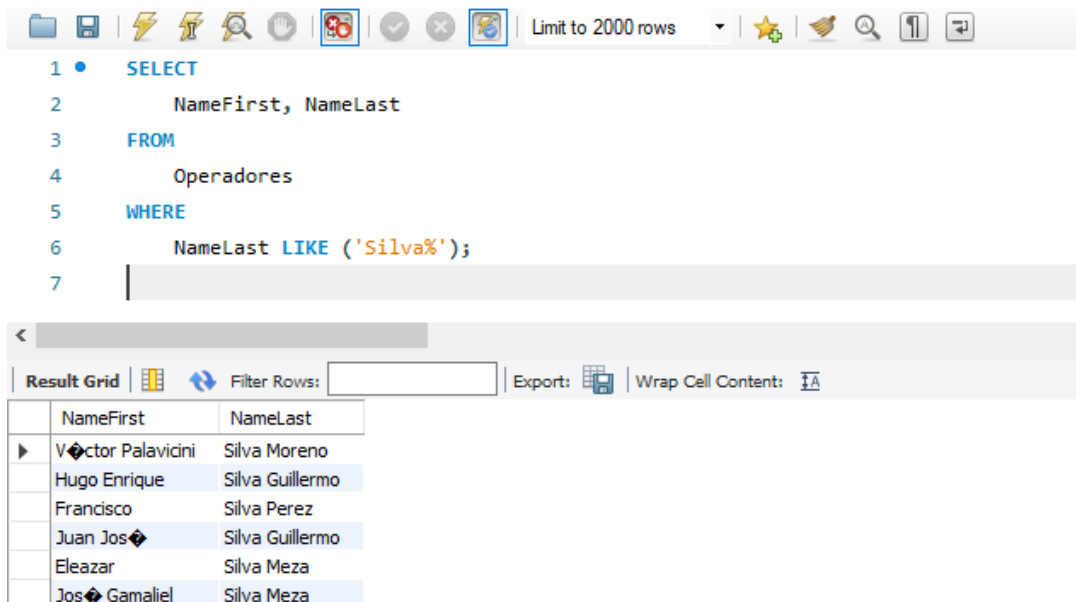
```
1 • SELECT
2     id, material_category, dump, load_count
3 FROM
4     Produccion
5 WHERE
6     material_category = 'ore'
7     AND dump = 'quebradora';
8
```

Below the query editor is the "Result Grid" showing the results of the query. The grid has four columns: id, material_category, dump, and load_count. The results are as follows:

	id	material_category	dump	load_count
▶	1	ore	QUEBRADORA	6
	5	ore	QUEBRADORA	4
	9	ore	QUEBRADORA	3
	13	ore	QUEBRADORA	6
	15	ore	QUEBRADORA	3
	18	ore	QUEBRADORA	1
	20	ore	QUEBRADORA	6
	21	ore	QUEBRADORA	4
	24	ore	QUEBRADORA	1

6. Dentro de la tabla Operadores, obtener los operadores con apellido Silva.

```
SELECT
    NameFirst, NameLast
FROM
    Operadores
WHERE
    NameLast LIKE ('Silva%');
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT
2     NameFirst, NameLast
3 FROM
4     Operadores
5 WHERE
6     NameLast LIKE ('Silva%');
7
```

Below the query editor is the "Result Grid" showing the results of the query. The grid has two columns: NameFirst and NameLast. The results are as follows:

	NameFirst	NameLast
▶	Véctor Palavicini	Silva Moreno
	Hugo Enrique	Silva Guillermo
	Francisco	Silva Perez
	Juan Jos	Silva Guillermo
	Eleazar	Silva Meza
	Jos Gamaliel	Silva Meza

7. Dentro de la tabla Operadores, obtener los operadores que su nombre termina con "el".

SELECT

NameFirst, NameLast

FROM

Operadores

WHERE

NameFirst LIKE ('%el');

The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 2000 rows' dropdown. The query editor displays the following SQL code:

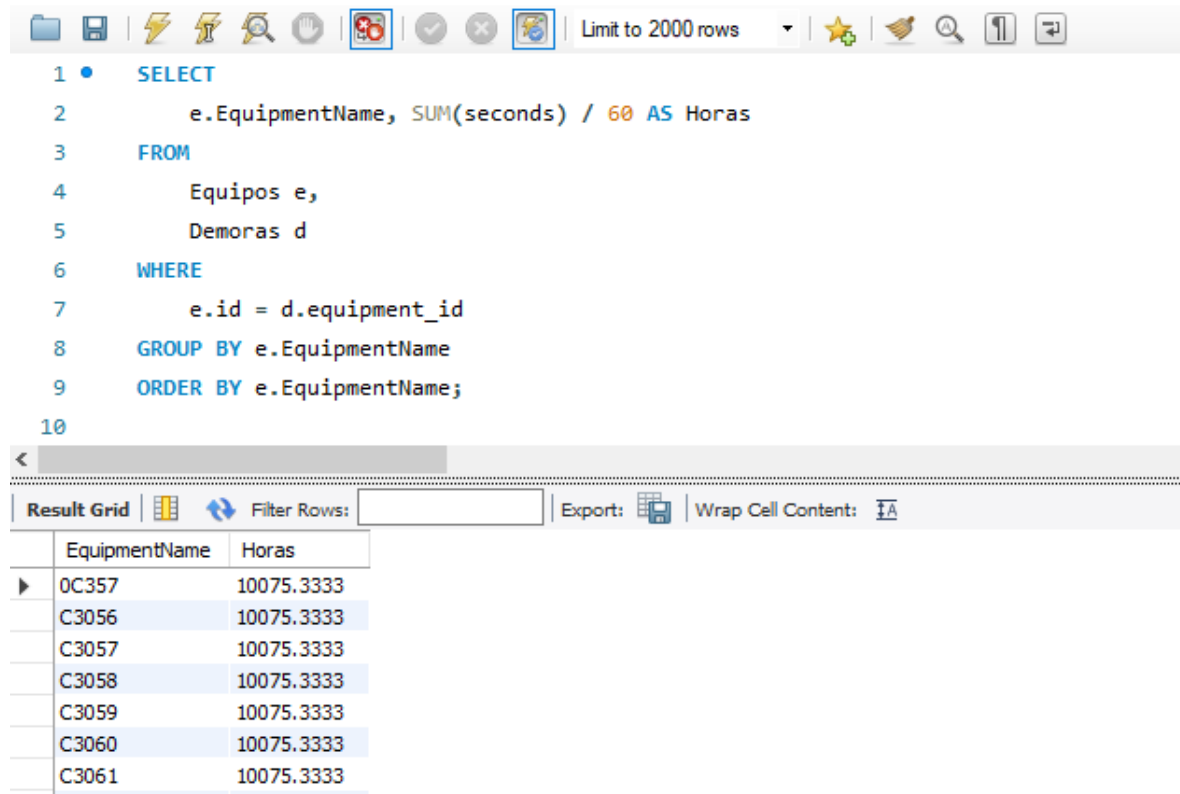
```
1 • SELECT
2     NameFirst, NameLast
3 FROM
4     Operadores
5 WHERE
6     NameFirst LIKE ('%el');
7
```

Below the query editor is the 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays the following data:

	NameFirst	NameLast
▶	Abel	Campos Martóñez
	Esquibel	Munguía Monroy
	Gabriel	Martóñez López
	Miguel Angel	Damián Vázquez
	Emmanuel	Lizardi Sandoval
	Manuel	Solis García
	Rafael	Vega González
	Manuel	Elizaga Curiel

8. Dentro de la tabla demoras, obtener las horas de demora con Código 939 por camión.

```
SELECT
    e.EquipmentName, SUM(seconds) / 60 AS Horas
FROM
    Equipos e,
    Demoras d
WHERE
    e.id = d.equipment_id
GROUP BY e.EquipmentName
ORDER BY e.EquipmentName;
```



The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

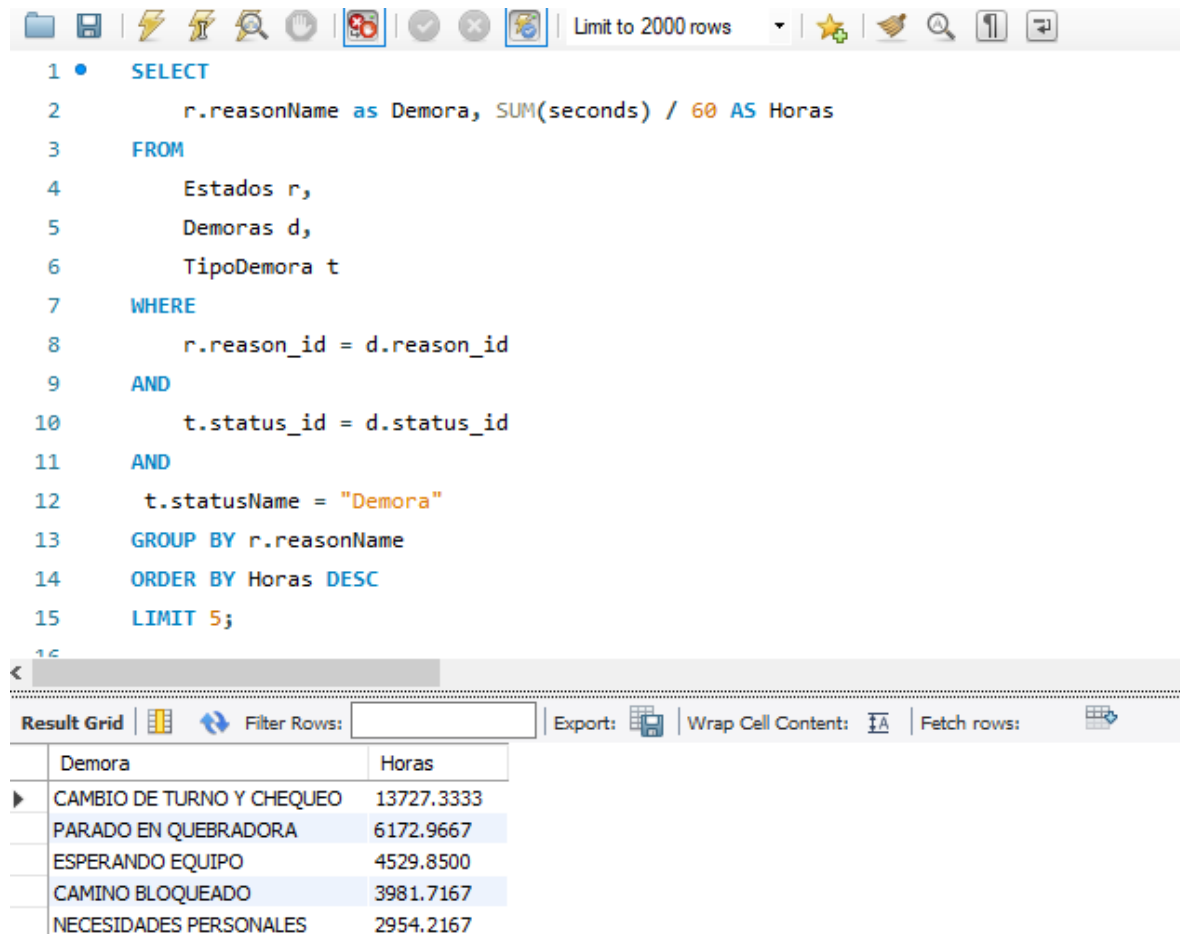
```
1 • SELECT
2     e.EquipmentName, SUM(seconds) / 60 AS Horas
3 FROM
4     Equipos e,
5     Demoras d
6 WHERE
7     e.id = d.equipment_id
8 GROUP BY e.EquipmentName
9 ORDER BY e.EquipmentName;
10
```

Below the query editor is the 'Result Grid' section. It has a toolbar with 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The results are displayed in a table with two columns: 'EquipmentName' and 'Horas'.

EquipmentName	Horas
0C357	10075.3333
C3056	10075.3333
C3057	10075.3333
C3058	10075.3333
C3059	10075.3333
C3060	10075.3333
C3061	10075.3333

9. Dentro de la tabla demoras, obtener el top 5 de demoras en total.

```
SELECT
    r.reasonName as Demora, SUM(seconds) / 60 AS Horas
FROM
    Estados r,
    Demoras d,
    TipoDemora t
WHERE
    r.reason_id = d.reason_id
AND
    t.status_id = d.status_id
AND
    t.statusName = "Demora"
GROUP BY r.reasonName
ORDER BY Horas DESC
LIMIT 5;
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

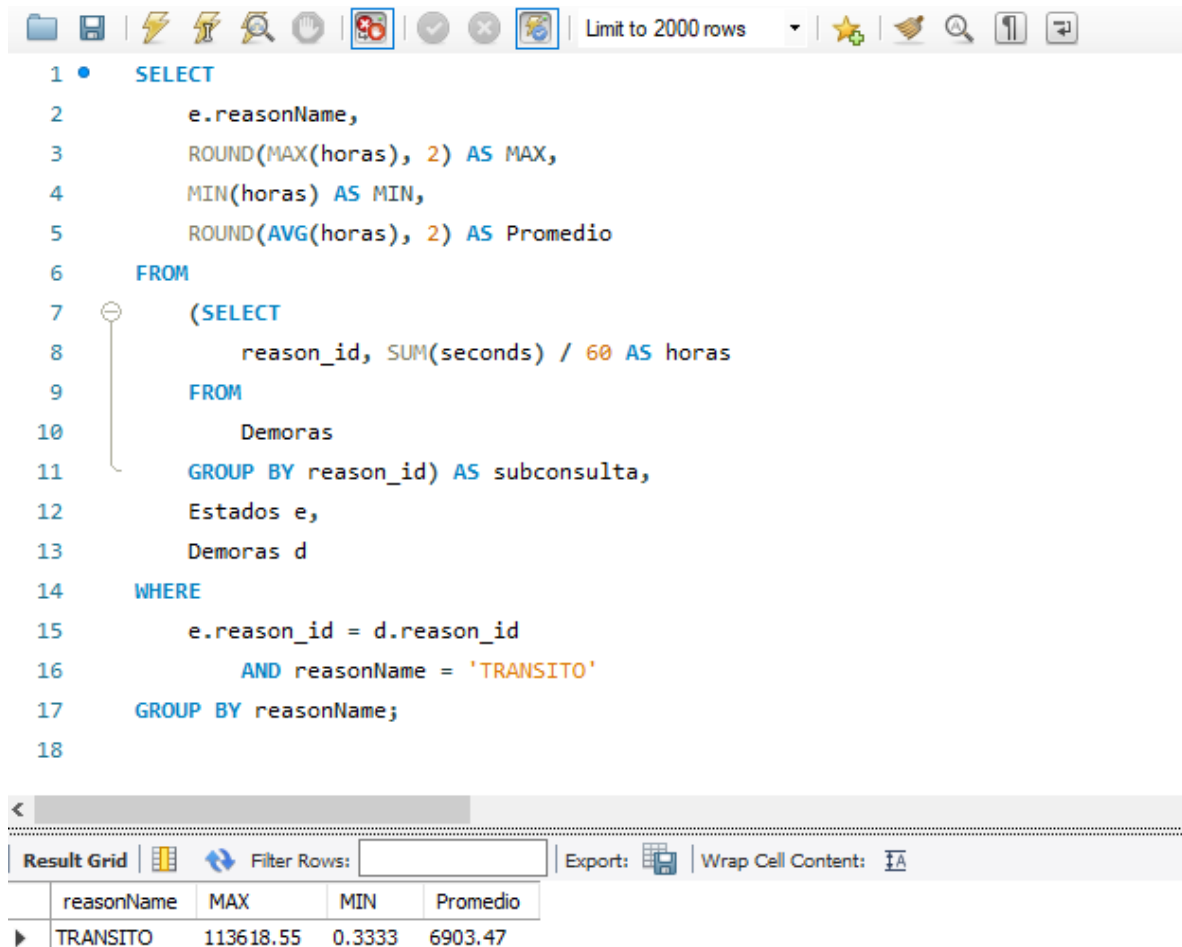
```
1 • SELECT
2     r.reasonName as Demora, SUM(seconds) / 60 AS Horas
3 FROM
4     Estados r,
5     Demoras d,
6     TipoDemora t
7 WHERE
8     r.reason_id = d.reason_id
9 AND
10    t.status_id = d.status_id
11 AND
12    t.statusName = "Demora"
13 GROUP BY r.reasonName
14 ORDER BY Horas DESC
15 LIMIT 5;
```

Below the query editor, the results are displayed in a table. The table has two columns: "Demora" and "Horas". The results are as follows:

Demora	Horas
CAMBIO DE TURNO Y CHEQUEO	13727.3333
PARADO EN QUEBRADORA	6172.9667
ESPERANDO EQUIPO	4529.8500
CAMINO BLOQUEADO	3981.7167
NECESIDADES PERSONALES	2954.2167

10. Obtén el máximo, mínimo y promedio de las horas de demora de "Transito".

```
SELECT
    e.reasonName,
    ROUND(MAX(horas), 2) AS MAX,
    MIN(horas) AS MIN,
    ROUND(AVG(horas), 2) AS Promedio
FROM
    (SELECT
        reason_id, SUM(seconds) / 60 AS horas
    FROM
        Demoras
    GROUP BY reason_id) AS subconsulta,
    Estados e,
    Demoras d
WHERE
    e.reason_id = d.reason_id
    AND reasonName = 'TRANSITO'
GROUP BY reasonName;
```

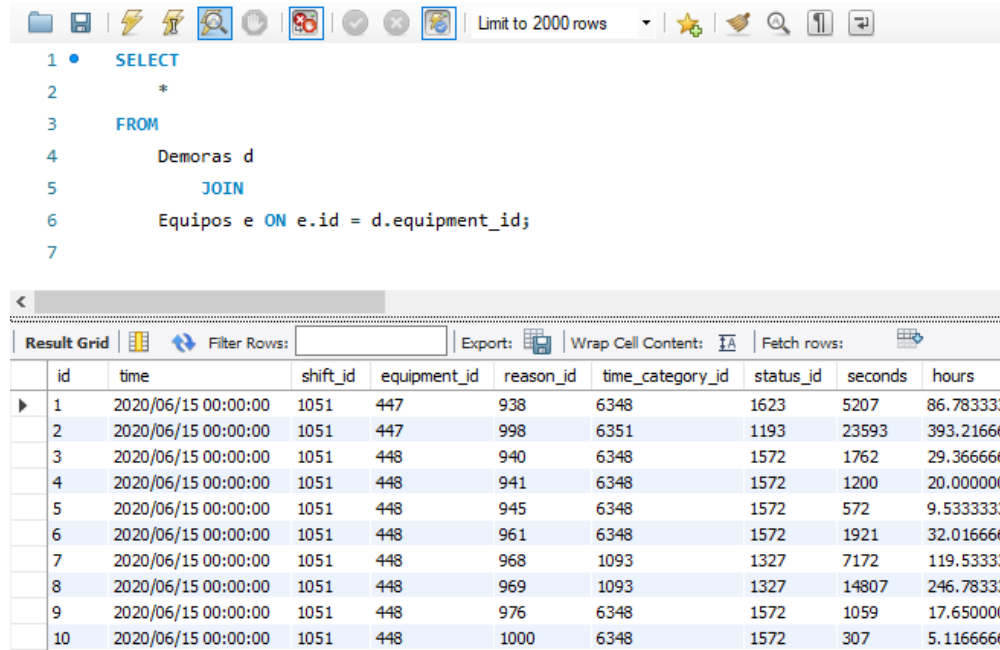


The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, search, and execution, along with a 'Limit to 2000 rows' dropdown. The SQL editor displays the query from the previous block, with line numbers 1 through 18. Below the editor, the 'Result Grid' is visible, showing the query results. The grid has columns for 'reasonName', 'MAX', 'MIN', and 'Promedio'. A single row of data is shown for 'TRANSITO'.

reasonName	MAX	MIN	Promedio
TRANSITO	113618.55	0.3333	6903.47

11. Muestra la información del camión desde la tabla demoras.

```
SELECT
*
FROM
Demoras d
JOIN
Equipos e ON e.id = d.equipment_id;
```



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

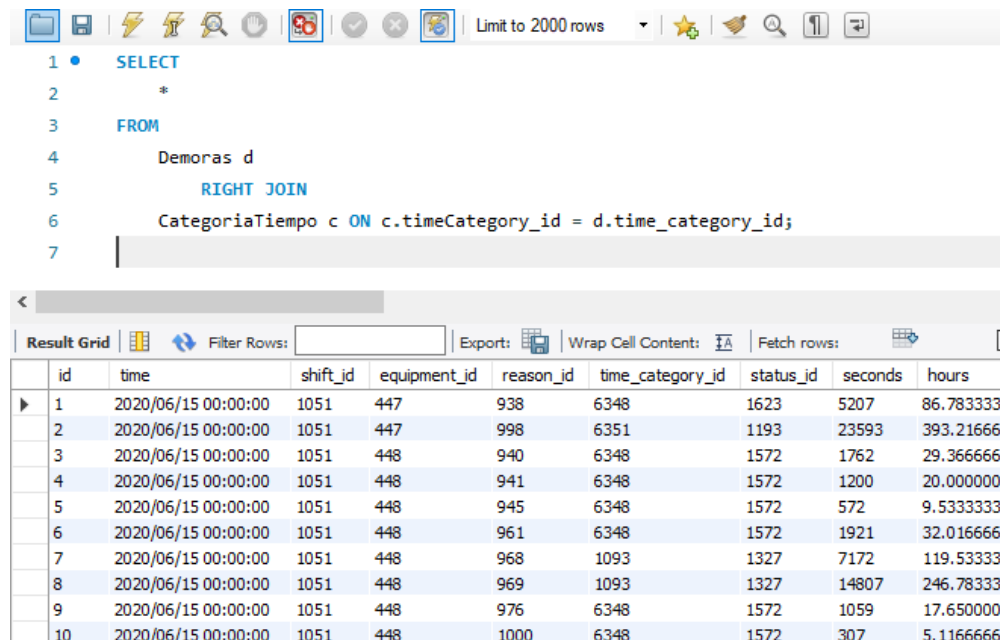
```
1 SELECT
2 *
3 FROM
4 Demoras d
5 JOIN
6 Equipos e ON e.id = d.equipment_id;
7
```

Below the query editor is the "Result Grid" showing 10 rows of data. The columns are: id, time, shift_id, equipment_id, reason_id, time_category_id, status_id, seconds, and hours.

	id	time	shift_id	equipment_id	reason_id	time_category_id	status_id	seconds	hours
▶	1	2020/06/15 00:00:00	1051	447	938	6348	1623	5207	86.783333
	2	2020/06/15 00:00:00	1051	447	998	6351	1193	23593	393.216666
	3	2020/06/15 00:00:00	1051	448	940	6348	1572	1762	29.366666
	4	2020/06/15 00:00:00	1051	448	941	6348	1572	1200	20.000000
	5	2020/06/15 00:00:00	1051	448	945	6348	1572	572	9.533333
	6	2020/06/15 00:00:00	1051	448	961	6348	1572	1921	32.016666
	7	2020/06/15 00:00:00	1051	448	968	1093	1327	7172	119.53333
	8	2020/06/15 00:00:00	1051	448	969	1093	1327	14807	246.78333
	9	2020/06/15 00:00:00	1051	448	976	6348	1572	1059	17.650000
	10	2020/06/15 00:00:00	1051	448	1000	6348	1572	307	5.116666

12. Muestra la información de categoría de tiempo desde la tabla demoras

```
SELECT
*
FROM
Demoras d
RIGHT JOIN
CategoriaTiempo c ON c.timeCategory_id = d.time_category_id;
```



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

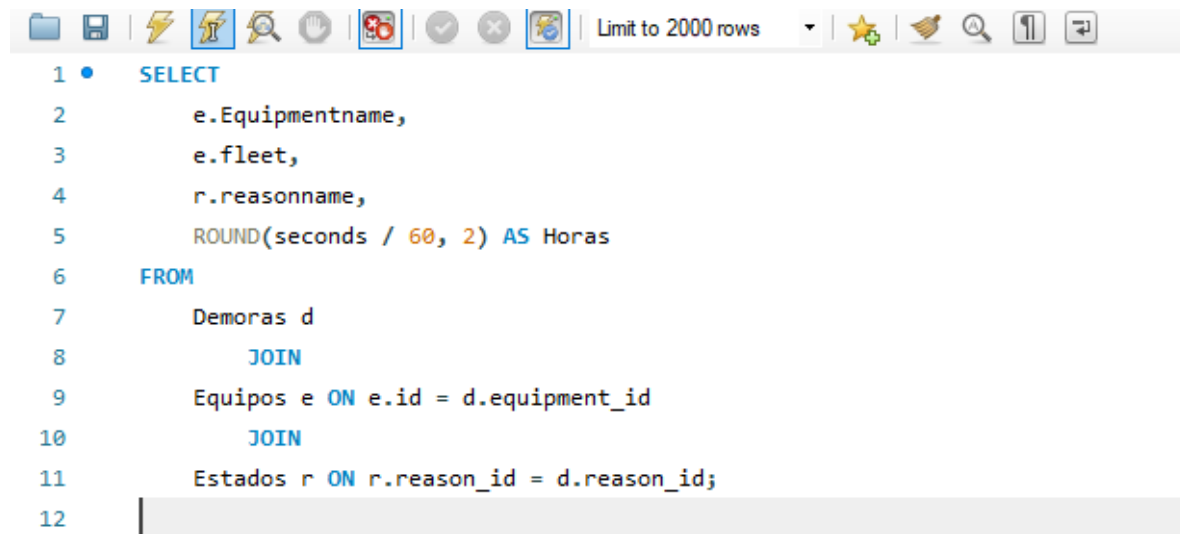
```
1 SELECT
2 *
3 FROM
4 Demoras d
5 RIGHT JOIN
6 CategoriaTiempo c ON c.timeCategory_id = d.time_category_id;
7
```

Below the query editor is the "Result Grid" showing 10 rows of data. The columns are: id, time, shift_id, equipment_id, reason_id, time_category_id, status_id, seconds, and hours.

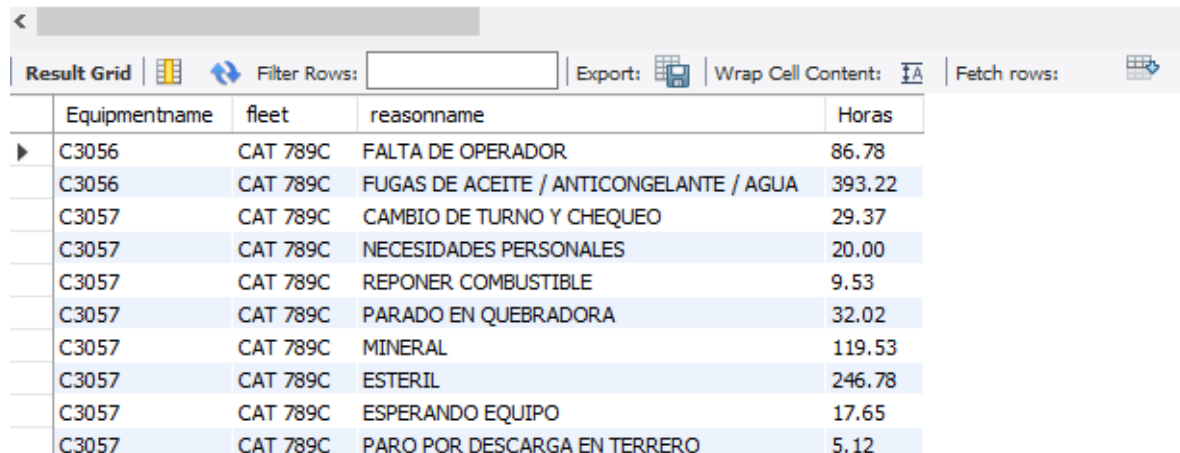
	id	time	shift_id	equipment_id	reason_id	time_category_id	status_id	seconds	hours
▶	1	2020/06/15 00:00:00	1051	447	938	6348	1623	5207	86.783333
	2	2020/06/15 00:00:00	1051	447	998	6351	1193	23593	393.216666
	3	2020/06/15 00:00:00	1051	448	940	6348	1572	1762	29.366666
	4	2020/06/15 00:00:00	1051	448	941	6348	1572	1200	20.000000
	5	2020/06/15 00:00:00	1051	448	945	6348	1572	572	9.533333
	6	2020/06/15 00:00:00	1051	448	961	6348	1572	1921	32.016666
	7	2020/06/15 00:00:00	1051	448	968	1093	1327	7172	119.53333
	8	2020/06/15 00:00:00	1051	448	969	1093	1327	14807	246.78333
	9	2020/06/15 00:00:00	1051	448	976	6348	1572	1059	17.650000
	10	2020/06/15 00:00:00	1051	448	1000	6348	1572	307	5.116666

13. Obtén el nombre del equipo, familia del equipo, nombre de demoras y horas de demora.

```
SELECT
    e.Equipmentname,
    e.fleet,
    r.reasonname,
    ROUND(seconds / 60, 2) AS Horas
FROM
    Demoras d
    JOIN
    Equipos e ON e.id = d.equipment_id
    JOIN
    Estados r ON r.reason_id = d.reason_id;
```



```
1 • SELECT
2     e.Equipmentname,
3     e.fleet,
4     r.reasonname,
5     ROUND(seconds / 60, 2) AS Horas
6 FROM
7     Demoras d
8     JOIN
9     Equipos e ON e.id = d.equipment_id
10    JOIN
11    Estados r ON r.reason_id = d.reason_id;
12
```



	Equipmentname	fleet	reasonname	Horas
▶	C3056	CAT 789C	FALTA DE OPERADOR	86.78
	C3056	CAT 789C	FUGAS DE ACEITE / ANTICONGELANTE / AGUA	393.22
	C3057	CAT 789C	CAMBIO DE TURNO Y CHEQUEO	29.37
	C3057	CAT 789C	NECESIDADES PERSONALES	20.00
	C3057	CAT 789C	REPONER COMBUSTIBLE	9.53
	C3057	CAT 789C	PARADO EN QUEBRADORA	32.02
	C3057	CAT 789C	MINERAL	119.53
	C3057	CAT 789C	ESTERIL	246.78
	C3057	CAT 789C	ESPERANDO EQUIPO	17.65
	C3057	CAT 789C	PARO POR DESCARGA EN TERRERO	5.12

14. Obtén una vista con el total de horas de alimentos por camión ordenando por el mayor.

create view TiempoAlimentosPorCamion as

(SELECT

 e.EquipmentName AS Camión,
 round(sum(hours),1) AS Horas

FROM

 Demoras d

 JOIN

 Equipos e ON e.id = d.equipment_id

 JOIN

 Estados r ON r.reason_id = d.reason_id

WHERE

 r.reasonname = 'ALIMENTOS'

group by e.EquipmentName

ORDER BY Horas DESC);

The screenshot shows a database management interface. On the left, a 'SCHEMAS' pane lists various databases and tables. The 'SVSchema' database is expanded, showing tables like 'CategoriaTiempo', 'Demoras', 'Equipos', 'Estados', 'Operadores', 'Produccion', 'TipoDemora', and 'Turno'. The 'TiempoAlimentosPorCa' view is highlighted. The main pane displays a SQL query: `SELECT * FROM SVSchema.TiempoAlimentosPorCamion;`. Below the query, a 'Result Grid' shows the output of the query, which is a list of trucks and their total hours for food. The results are ordered by hours in descending order.

Camión	Horas
C3057	580.1
C3069	405.8
C3071	337.9
C3068	328.5
C3067	311.0
C3075	306.7
C3064	252.4
C3074	224.1
C3081	212.1
C3077	194.9
C3082	178.0
C3083	171.7
C3061	163.6
C3063	133.0
C3065	117.7
C3062	111.3
C3059	110.7
C3079	102.6
C3076	91.5
C3056	68.3
C3058	63.2

15. Obtén una vista con la producción por camión CAT 793F donde incluya nombre de camión, Operador y tipo de material.

create view ProduccionCamion as

```
(SELECT
    p.time,
    t.Shift AS Turno,
    e.EquipmentName AS Camion,
    o.EquipmentOperatorName AS Operador,
    p.material_category AS TipoMaterial,
    sum(p.material_tonnage) AS Tonelaje
FROM
    Produccion p
    JOIN
    Equipos e ON e.id = p.truck_id
    JOIN
    Operadores o ON o.operadores_id = p.truck_operator_id
    JOIN
    Turno t on t.shift_id = p.shift_id
WHERE
    e.fleet = 'CAT 793F'
GROUP BY time, Turno, Camion, Operador, TipoMaterial
order by time);
```

SCHEMAS

Filter objects

- ETOR
- ETOR2
- HE
- JDA
- JDA_world
- JV
- MET
- Modulos2
- MovieRatings
- PRUEBA
- PRUEBA1
- prueba2
- sakila
- Seguimiento_Requets
- SV
- SVSchema**
 - Tables
 - CategoriaTiempo
 - Demoras
 - Equipos
 - Estados
 - Operadores
 - Produccion
 - TipoDemora
 - Turno
 - Views
 - ProduccionCamion**
 - TiempoAlimentosPorCa
 - Stored Procedures

1 • SELECT * FROM SVSchema.ProduccionCamion;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [f1](#)

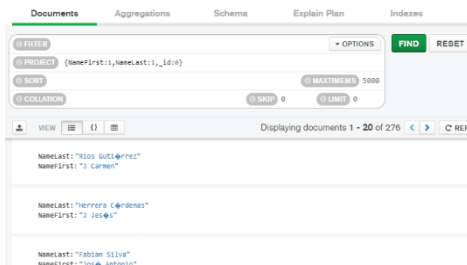
time	Turno	Camion	Operador	TipoMaterial	Tonelaje
2020/06/15 00:00:00	Diurno	C3074	GONZALEZ DE LOS SANTOS Omar	waste	2870
2020/06/15 00:00:00	Diurno	C3075	HERNANDEZ LORENZO Marcos Eloy	waste	820
2020/06/15 00:00:00	Diurno	C3075	VERA GARCIA Efraon	ore	800
2020/06/15 00:00:00	Diurno	C3075	VERA GARCIA Efraon	waste	1025
2020/06/15 00:00:00	Diurno	C3077	HERNANDEZ LORENZO Marcos Eloy	waste	615
2020/06/15 00:00:00	Diurno	C3079	CERVANTES VIZQUEZ Mario	ore	400
2020/06/15 00:00:00	Diurno	C3080	MUNGUIA ROMERO Carlos No	ore	2200
2020/06/15 00:00:00	Diurno	C3081	GUTIERREZ CUIEL Rigoberto	ore	600
2020/06/15 00:00:00	Diurno	C3081	GUTIERREZ CUIEL Rigoberto	waste	2870
2020/06/15 00:00:00	Diurno	C3082	MICHEL DIAZ Francisco Javier	ore	200
2020/06/15 00:00:00	Diurno	C3082	MICHEL DIAZ Francisco Javier	waste	3075
2020/06/15 00:00:00	Diurno	C3083	CEJA CUELLAR Jos	waste	1025
2020/06/15 00:00:00	Mixto	C3074	MICHEL CAMPOS Juan Jos	ore	600
2020/06/15 00:00:00	Mixto	C3074	MICHEL CAMPOS Juan Jos	waste	1435
2020/06/15 00:00:00	Mixto	C3075	LEZARO MANRIQUEZ Omar Israel	waste	2665
2020/06/15 00:00:00	Mixto	C3079	LECONA PEREZ Leonardo Guadalupe	ore	400
2020/06/15 00:00:00	Mixto	C3079	LECONA PEREZ Leonardo Guadalupe	waste	2050
2020/06/15 00:00:00	Mixto	C3080	MUNGUIA ROMERO Carlos No	ore	600

MongoDB Compass

1. Dentro de la tabla Operadores, obtén el nombre y apellido de todos los Operadores.

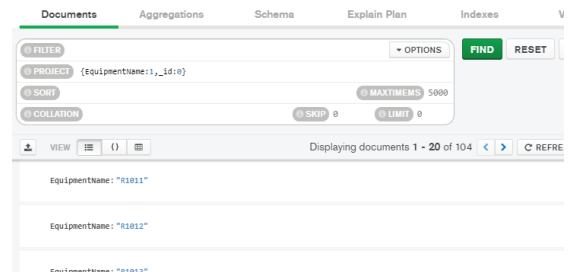
```
{
  project: {
    NameFirst: 1,
    NameLast: 1,
    _id: 0
  }
}
```

SalvadorVizcaino.Operadores



2. Dentro de la tabla Equipos, obtienen los equipos de acarreo.

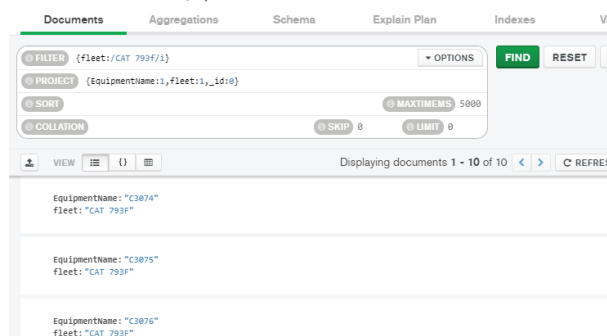
```
{
  project: {
    EquipmentName: 1,
    _id: 0
  }
}
```



3 Dentro de la tabla Equipos, obtén los camiones modelo CAT 793F.

```
{
  filter: {
    fleet: 'CAT 793F'
  },
  project: {
    EquipmentName: 1,
    fleet: 1,
    _id: 0
  }
}
```

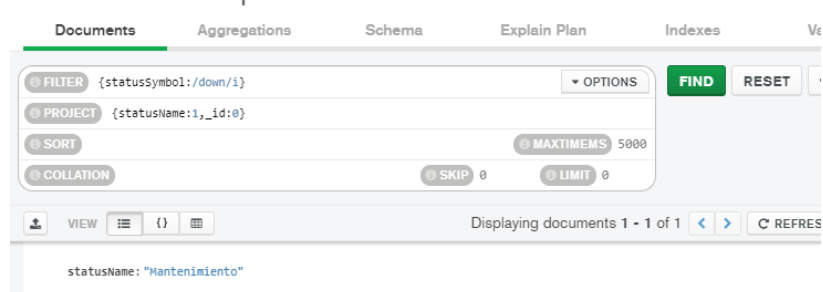
SalvadorVizcaino.Equipos



4 Cual es el nombre del estado el cual tiene símbolo "Down"?

```
{
  filter: {
    statusSymbol: 'down'
  },
  project: {
    statusName: 1,
    _id: 0
  }
}
```

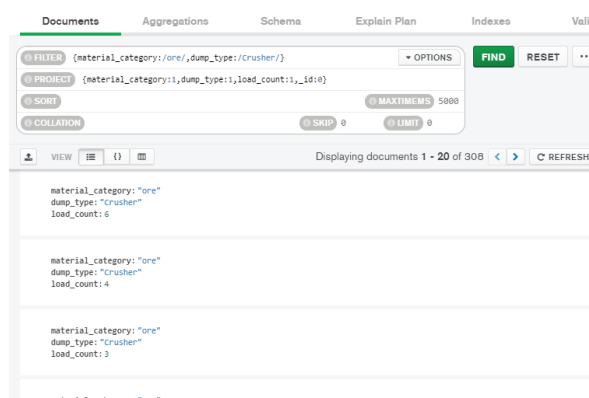
SalvadorVizcaino.TipoDemora



5 Dentro de la tabla Producción, obtén los viajes de mineral que fueron a la trituradora.

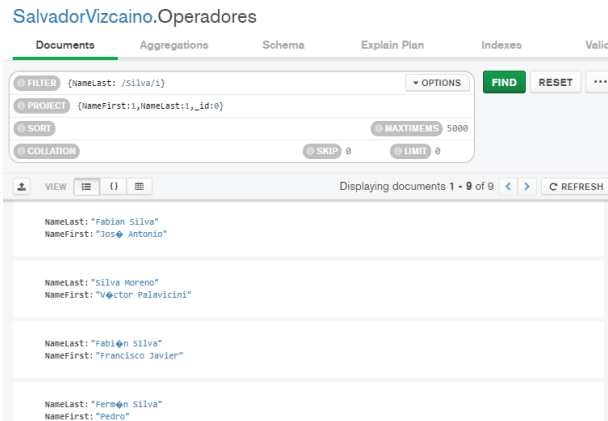
```
{
  filter: {
    material_category: 'ore',
    dump_type: 'Crusher'
  },
  project: {
    material_category: 1,
    dump_type: 1,
    load_count: 1
  }
}
```

SalvadorVizcaino.Produccion



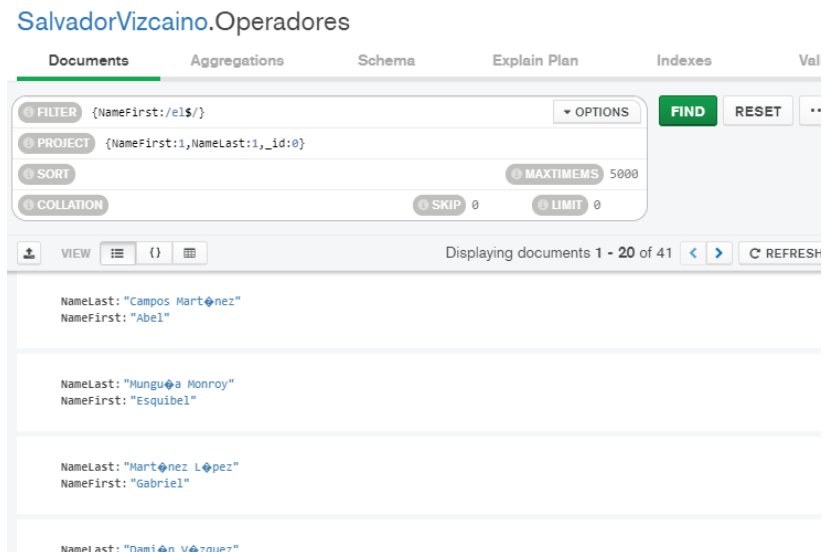
6. Dentro de la tabla Operadores, obtén los operadores con apellido Silva.

```
{
  filter: {
    NameLast: RegExp('Silva', i)
  },
  project: {
    NameFirst: 1,
    NameLast: 1,
    _id: 0
  }
}
```



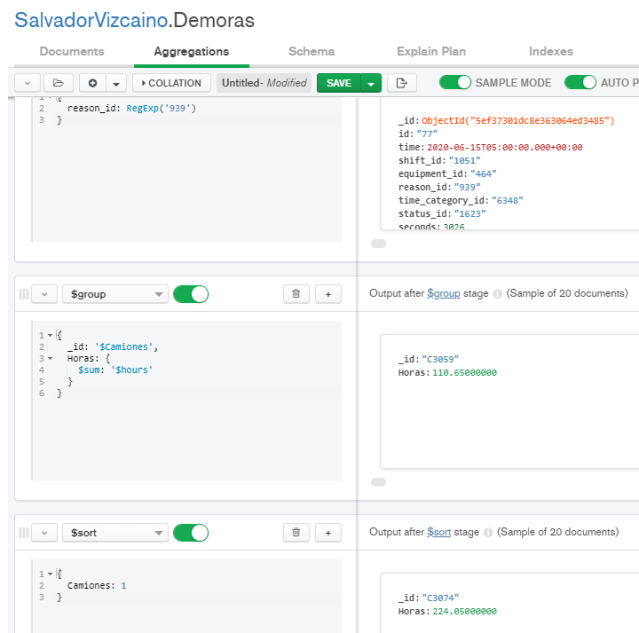
7. Dentro de la tabla Operadores, obtén los operadores que su nombre termina con "el".

```
{
  filter: {
    NameFirst: RegExp('el$', i)
  },
  project: {
    NameFirst: 1,
    NameLast: 1,
    _id: 0
  }
}
```



8. Dentro de la tabla demoras, obtener las horas de demora con código 939 por camión.

```
{ $lookup: {
  from: 'Equipos',
  localField: 'equipment_id',
  foreignField: 'id',
  as: 'Camiones2'
}}, { $addFields: {
  Camiones1: { $arrayElemAt: ["$Camiones2", 0]
}}, { $addFields: {
  Camiones: "$Camiones1.EquipmentName"
}}, { $match: {
  reason_id: /939/
}}, { $group: {
  _id: "$Camiones",
  Horas: {
    $sum: "$hours"
  }
}}, { $sort: {
  Camiones: 1
}}
```



9. Dentro de la tabla demoras, obtén el top 5 de demoras en total.

```
[{$lookup: {
  from: 'Estados',
  localField: 'reason_id',
  foreignField: 'reason_id',
  as: 'Estado2'
}}, {$lookup: {
  from: 'TipoDemora',
  localField: 'status_id',
  foreignField: 'status_id',
  as: 'Tipo_Demora2'
}}, {$addFields: {
  Estado1: {$arrayElemAt: ["$Estado2",0]}
}}, {$addFields: {
  Tipo_Demora1: {$arrayElemAt: ["$Tipo_Demora2",0]}
}}, {$addFields: {
  Estado: "$Estado1.reasonName"
}}, {$addFields: {
  Tipo_Demora: "$Tipo_Demora1.statusName"
}}, {$match: {
  Tipo_Demora: /Demora/
}}, {$project: {
  _id:0,Estado:1,hours:1
}}, {$group: {
  _id: "$Estado",
  Hora: {
    $sum: "$hours"
  }
}}, {$sort: {
  Hora: -1
}}, {$limit: 5}]
```

SalvadorVizcaino.Demoras

The screenshot shows the MongoDB Aggregations Builder interface for a collection named 'Demoras'. The pipeline consists of three stages: \$group, \$sort, and \$limit. The \$group stage groups documents by 'Estado' and calculates the sum of 'hours' as '\$sum: \$hours'. The \$sort stage sorts the results by 'Hora' in descending order (-1). The \$limit stage limits the results to the top 5 documents. The interface shows the output of each stage, with the final output being a sample of 5 documents.

Stage 1: \$group (Output after \$group stage: Sample of 13 documents)

```
1 {
2   _id: '$Estado',
3   Hora: {
4     $sum: '$hours'
5   }
6 }
```

Stage 2: \$sort (Output after \$sort stage: Sample of 13 documents)

```
1 {
2   Hora: -1
3 }
```

Stage 3: \$limit (Output after \$limit stage: Sample of 5 documents)

```
1 5
```

10 Obten el máximo, mínimo y promedio de las horas de demora de "Transito".

```
{
  $lookup: {
    from: 'Estados',
    localField: 'reason_id',
    foreignField: 'reason_id',
    as: 'Estado2'
  },
  $lookup: {
    from: 'TipoDemora',
    localField: 'status_id',
    foreignField: 'status_id',
    as: 'Tipo_Demora2'
  },
  $addFields: {
    Estado1: {$arrayElemAt: ["$Estado2", 0]}
  },
  $addFields: {
    Tipo_Demora1: {$arrayElemAt: ["$Tipo_Demora2", 0]}
  },
  $addFields: {
    Estado: "$Estado1.reasonName"
  },
  $addFields: {
    Tipo_Demora: "$Tipo_Demora1.statusName"
  },
  $match: {
    Estado: /Transito/i
  },
  $project: {
    _id: 0, Estado: 1, Tipo_Demora: 1, hours: 1
  },
  $match: {
    Estado: /Transito/i
  },
  $group: {
    _id: "$Estado",
    Maximo: {$max: "$hours"},
    Mínimo: {$min: "$hours"},
    Promedio: {$avg: "$hours"}
  }
}
```

SalvadorVizcaino.Demoras

The screenshot shows the MongoDB Aggregations Builder interface. The pipeline is as follows:

- \$lookup**: Joins 'Estados' to 'TipoDemora' based on 'reason_id' and 'status_id'.
- \$addFields**: Adds 'Estado1' and 'Tipo_Demora1' to the document.
- \$addFields**: Adds 'Estado' and 'Tipo_Demora' to the document.
- \$match**: Filters documents where 'Estado' is 'Transito'.
- \$group**: Groups documents by 'Estado' and calculates 'Maximo', 'Mínimo', and 'Promedio' for 'hours'.

The output after the \$group stage shows:

```
{
  "_id": "TRANSITO",
  "Maximo": 282.05,
  "Mínimo": 1.566666667,
  "Promedio": 23.9185555551
}
```

11 Muestra la información del camión desde la tabla demoras.

```
{
  $lookup: {
    from: 'Equipos',
    localField: 'equipment_id',
    foreignField: 'id',
    as: 'Camion2'
  },
  $addFields: {
    Camion1: {$arrayElemAt: ["$Camion2", 0]}
  },
  $addFields: {
    camión: "$Camion1.EquipmentName"
  },
  $addFields: {
    Flota: "$Camion1.fleet"
  },
  $addFields: {
    Tipo: "$Camion1.type"
  },
  $project: {
    Camion2: 0, Camion1: 0
  }
}
```

SalvadorVizcaino.Demoras

The screenshot shows the MongoDB Aggregations Builder interface. The pipeline is as follows:

- \$lookup**: Joins 'Equipos' to 'Camion2' based on 'equipment_id' and 'id'.
- \$addFields**: Adds 'Camion1' to the document.
- \$addFields**: Adds 'camión', 'Flota', and 'Tipo' to the document.
- \$project**: Projects the document with 'Camion2' and 'Camion1' set to 0.

The output after the \$project stage shows:

```
{
  "equipment_id": "447",
  "reason_id": "938",
  "time_category_id": "6348",
  "status_id": "1623",
  "seconds": 5287,
  "hours": 86.78333333,
  "camión": "CAT 789C",
  "Flota": "CAT 789C",
  "Tipo": "Truck"
}
```


12 Muestra la información de categoría de tiempo desde la tabla demoras.

```
{
  $lookup: {
    from: 'CategoriaTiempo',
    localField: 'time_category_id',
    foreignField: 'timeCategory_id',
    as: 'timeCategory2'
  },
  $addFields: {
    timeCategory1: {
      $arrayElemAt: ["$timeCategory2", 0]
    },
    $addFields: {
      timeCategory: "$timeCategory1.timeCategoryName"
    },
    $addFields: {
      Symbol: "$timeCategory1.timeCategorySymbol"
    },
    $project: {
      timeCategory2: 0,
      timeCategory1: 0
    }
  }
}
```

SalvadorVizcaino.Demoras

Documents Aggregations Schema Explain Plan Indexes Validate

COLLATION Untitled SAVE SAMPLE MODE AUTO PREVIEW

timeCategory: '\$timeCategory1.timeCategoryName'

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

```
{
  "_id": "ObjectId('5ef37301dc8e363064ed3439')",
  "id": "1",
  "time": "2020-06-15T05:00:00.000+00:00",
  "shift_id": "1051",
  "equipment_id": "447",
  "reason_id": "938",
  "time_category_id": "6348",
  "status_id": "1623",
  "seconds": 5207
}
```

\$addFields

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

```
{
  "Symbol": '$timeCategory1.timeCategorySymbol'
}
```

\$project

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

```
{
  "timeCategory2": 0,
  "timeCategory1": 0
}
```

```
{
  "shift_id": "1051",
  "equipment_id": "447",
  "reason_id": "938",
  "time_category_id": "6348",
  "status_id": "1623",
  "seconds": 5207,
  "hours": 86.78333333,
  "timeCategory": "Demora",
  "Symbol": "delay"
}
```

13 Obtén el nombre del equipo, familia del equipo, nombre de demoras y horas de demora.

```

[{$lookup: {
  from: 'Equipos',
  localField: 'equipment_id',
  foreignField: 'id',
  as: 'Equipo2'
}}, {$addFields: {
  Equipo1: {$arrayElemAt:["$Equipo2",0]}
}}, {$addFields: {
  Equipo: "$Equipo1.EquipmentName",
  Familia: "$Equipo1.fleet"
}
}, {$lookup: {
  from: 'Estados',
  localField: 'reason_id',
  foreignField: 'reason_id',
  as: 'Estado2'
}}, {$addFields: {
  Estado1: {$arrayElemAt:["$Estado2",0]}
}}, {$addFields: {
  Demora: "$Estado1.reasonName"
}}, {$lookup: {
  from: 'CategoriaTiempo',
  localField: 'time_category_id',
  foreignField: 'timeCategory_id',
  as: 'timeCategory2'
}}, {$addFields: {
  timeCategory1: {$arrayElemAt:["$timeCategory2",0]}
}}, {$addFields: {
  timeCategory: "$timeCategory1.timeCategoryName"
}}, {$match: {
  timeCategory: /Demora/
}}, {$project: {
  _id:0,Equipo:1,Familia:1,Demora:1,hours:1
}}

```

SalvadorVizcaino.Demoras

Documents Aggregations Schema Explain Plan Indexes Valid

COLLATION Untitled SAVE SAMPLE MODE AUTO PREVIEW

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

```

1 {
2   timeCategory: 'timeCategory1.timeCategoryName'
3 }

```

```

_id: ObjectId("5ef37381dc8e363064ed3439")
id: "1"
time: 2020-06-15T05:00:00.000+00:00
shift_id: "1051"
equipment_id: "447"
reason_id: "938"
time_category_id: "6348"
status_id: "1623"
seconds: 5207

```

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

```

1 {
2   timeCategory: RegExp('Demora')
3 }

```

```

_id: ObjectId("5ef37381dc8e363064ed3439")
id: "1"
time: 2020-06-15T05:00:00.000+00:00
shift_id: "1051"
equipment_id: "447"
reason_id: "938"
time_category_id: "6348"
status_id: "1623"
seconds: 5207

```

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

```

1 {
2   _id: 0,
3   Equipo: 1,
4   Familia: 1,
5   Demora: 1,
6   hours: 1
7 }

```

```

hours: 86.78333333
Equipo: "C3056"
Familia: "CAT 789C"
Demora: "FALTA DE OPERADOR"

```

14 Obtener una vista con el total de horas de alimentos por camión ordenando por el mayor.

```
[{$lookup: {
  from: 'Estados',
  localField: 'reason_id',
  foreignField: 'reason_id',
  as: 'Demoras2'
}}, {$addFields: {
  Demoras1: {$arrayElemAt:["$Demoras2",0]}
}}, {$addFields: {
  Demoras: "$Demoras1.reasonName"
}}, {$match: {
  Demoras: /Alimentos/i
}}, {$lookup: {
  from: 'Equipos',
  localField: 'equipment_id',
  foreignField: 'id',
  as: 'Equipos2'
}}, {$addFields: {
  Equipos1: {$arrayElemAt:["$Equipos2",0]}
}}, {$addFields: {
  Equipos: "$Equipos1.EquipmentName"
}}, {$group: {
  _id: "$Equipos",
  Horas: {
    $sum: "$hours"
  }
}}, {$sort: {
  Horas: -1
}}]
```

SalvadorVizcaino.Demoras

Documents Aggregations Schema Explain Plan Indexes Validate

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

Equipos: '\$Equipos1.EquipmentName'

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

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

15 Obtener una vista con la producción por camión CAT 793F donde incluya nombre de camión, Operador y tipo de material.

```
{
  $lookup: {
    from: 'Operadores',
    localField: 'truck_operator_id',
    foreignField: 'operadores_id',
    as: 'Operadores2'
  },
  $addFields: {
    Operadores1: { $arrayElemAt: ['$Operadores2', 0] }
  },
  $addFields: {
    Operadores: "$Operadores1.EquipmentOperatorName"
  },
  $lookup: {
    from: 'Turno',
    localField: 'shift_id',
    foreignField: 'id',
    as: 'Turno2'
  },
  $addFields: {
    Turno1: { $arrayElemAt: ['$Turno2', 0] }
  },
  $addFields: {
    Turno: "$Turno1.Shift"
  },
  $lookup: {
    from: 'Equipos',
    localField: 'truck_id',
    foreignField: 'id',
    as: 'Camiones2'
  },
  $addFields: {
    Camiones1: { $arrayElemAt: ['$Camiones2', 0] }
  },
  $addFields: {
    Flota: "$Camiones1.fleet"
  },
  $match: {
    Flota: '/CAT 793F/i'
  },
  $project: {
    _id: 0,
    time: 1,
    Turno: 1,
    truck: 1,
    Flota: 1,
    Operadores: 1,
    material_category: 1,
    material_tonnage: 1
  }
}
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Filter your data' sidebar lists various collections, with 'ProduccionCamion F...' selected. The main panel displays the query results in a table format. The table has columns for 'time', 'truck', 'material_category', 'material_tonnage', 'Operadores', 'Turno', and 'Flota'. The results show four documents, each representing a production record for a specific truck and operator.

time	truck	material_category	material_tonnage	Operadores	Turno	Flota
2020-06-15T05:00:00.000+00:00	"C3082"	"waste"	2665	"MICHEL DAZ Francisco Javier"	"Diurno"	"CAT 793F"
2020-06-15T05:00:00.000+00:00	"C3080"	"ore"	1600	"HUNGU A ROMERO Carlos No"	"Diurno"	"CAT 793F"
2020-06-15T05:00:00.000+00:00	"C3075"	"waste"	205	"VERA GARC A Efraon"	"Diurno"	"CAT 793F"
2020-06-15T05:00:00.000+00:00	"C3082"	"ore"	200	"MICHEL DAZ Francisco Javier"	"Diurno"	"CAT 793F"