



# Carga bruta

Josmar Gustavo Palomino Castelán

Creación de la tabla que contendrá los datos de manera bruta.

The screenshot shows the MySQL Workbench interface. In the left sidebar, the 'Schemas' panel is open, showing a tree view of databases including 'practica' and 'tintoreria\_aries'. The 'tintoreria\_aries' database is selected, and the 'Tables' folder is expanded. The main editor window displays a SQL script to create a table named 'carga\_bruta' with the following columns: 'Folio\_Nota' (varchar(3), null), 'Telefono\_Cliente' (varchar(10), NOT NULL), 'Tipo\_Servicio' (varchar(15), NOT NULL), 'Fecha\_Entrega\_Estimada' (date, NOT NULL), 'Hora\_Entrega\_Estimada' (time, NOT NULL), 'Estatus' (varchar(15), NOT NULL), and 'Cargado' (varchar(15), null). The script is as follows:

```
CREATE TABLE carga_bruta (  
  Folio_Nota varchar(3) null,  
  Telefono_Cliente varchar(10) NOT NULL,  
  Tipo_Servicio varchar(15) NOT NULL,  
  Fecha_Entrega_Estimada date NOT NULL,  
  Hora_Entrega_Estimada time NOT NULL,  
  Estatus varchar(15) NOT NULL,  
  Cargado varchar(15) null  
);
```

Below the script, the 'Result Grid' shows the execution results of the SQL script. The 'Output' tab is selected, showing the following actions:

#	Time	Action	Message	Duration / Fetch
537	08:09:38	drop table carga_bruta	0 row(s) affected	0.093 sec
538	08:09:44	CREATE TABLE carga_bruta ( Folio_Nota varchar(3) null, Telefono_Cliente var...	0 row(s) affected	0.063 sec
539	08:09:49	select * from carga_bruta LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec

Cargamos la tabla carga\_bruta con los datos iguales a como se tiene en un documento Excel por medio comandos. (con clave nula, clave con otro carácter, y otro dato el cual ya existe en la tabla de notas la cual se le lleva a cabo la carga.

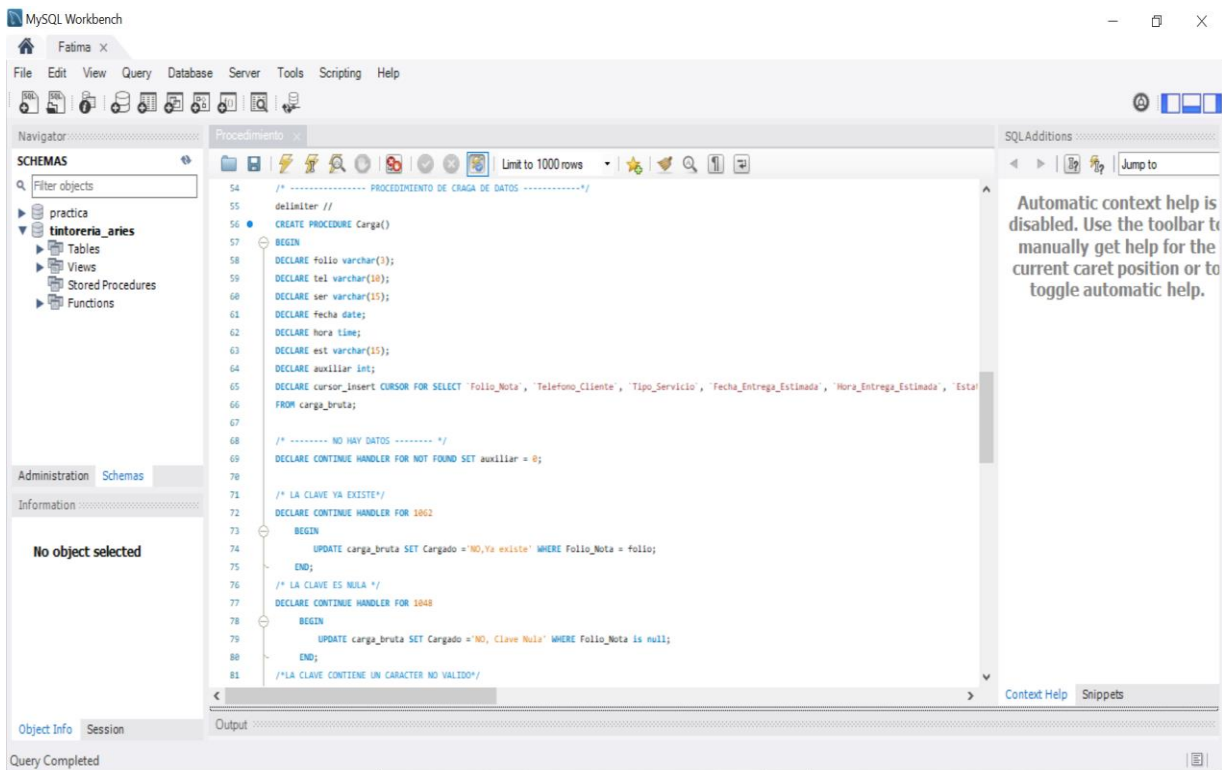
The screenshot shows the MySQL Workbench interface. In the left sidebar, the 'Schemas' panel is open, showing a tree view of databases including 'practica' and 'tintoreria\_aries'. The 'tintoreria\_aries' database is selected, and the 'Tables' folder is expanded. The main editor window displays a SQL script to insert data into the 'carga\_bruta' table. The script is as follows:

```
values (828, 5545303785, 'Reparacion', '2021/02/16', '19:30:00', '');  
insert into carga_bruta (folio_Nota, Telefono_Cliente, Tipo_Servicio, Fecha_Entrega_Estimada, Hora_Entrega_Estimada, Estatus  
values (829, 5596842072, 'Teñido', '2021/02/17', '19:30:00', '');  
insert into carga_bruta (folio_Nota, Telefono_Cliente, Tipo_Servicio, Fecha_Entrega_Estimada, Hora_Entrega_Estimada, Estatus
```

Below the script, the 'Result Grid' shows the execution results of the SQL script. The 'Output' tab is selected, showing the following actions:

#	Time	Action	Message	Duration / Fetch
548	08:14:00	insert into carga_bruta (folio_Nota, Telefono_Cliente, Tipo_Servicio, Fecha_Entreg...	1 row(s) affected	0.000 sec
549	08:14:04	insert into carga_bruta (folio_Nota, Telefono_Cliente, Tipo_Servicio, Fecha_Entreg...	1 row(s) affected	0.000 sec
550	08:14:07	select * from carga_bruta LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

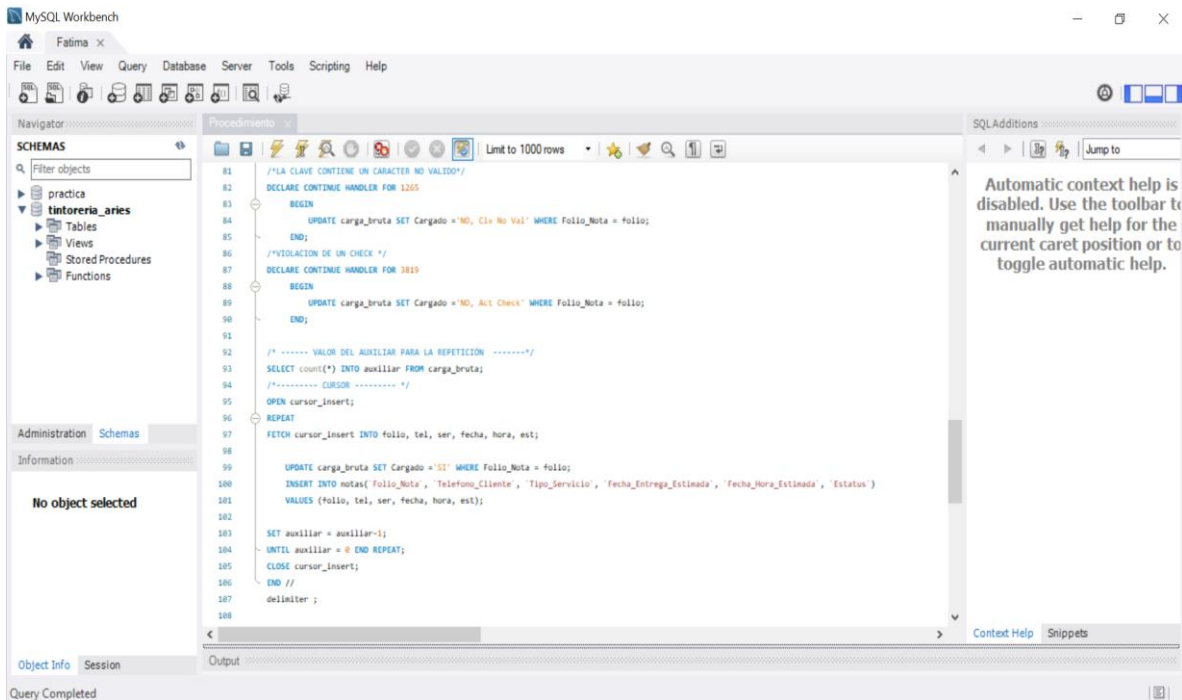
Creación de procedimiento para la carga de datos automática, da detalle si se registra o no un registro en la tabla de carga bruta.



The screenshot shows the MySQL Workbench interface with a stored procedure named 'Carga()' being created. The procedure is designed to handle data loading into a table named 'carga\_bruta'. It includes logic to check if a record already exists and to handle null values.

```
54 /* ----- PROCEDIMIENTO DE CARGA DE DATOS ----- */
55 delimiter //
56 CREATE PROCEDURE Carga()
57 BEGIN
58     DECLARE folio varchar(11);
59     DECLARE tel varchar(10);
60     DECLARE ser varchar(15);
61     DECLARE fecha date;
62     DECLARE hora time;
63     DECLARE est varchar(15);
64     DECLARE auxiliar int;
65     DECLARE cursor_insert CURSOR FOR SELECT 'Folio_Nota', 'Telefono_Cliente', 'Tipo_Servicio', 'Fecha_Entrega_Estimada', 'Hora_Entrega_Estimada', 'Estado'
66     FROM carga_bruta;
67
68 /* ----- NO HAY DATOS ----- */
69 DECLARE CONTINUE HANDLER FOR NOT FOUND SET auxiliar = 0;
70
71 /* LA CLAVE YA EXISTE */
72 DECLARE CONTINUE HANDLER FOR 1062
73 BEGIN
74     UPDATE carga_bruta SET Cargado = 'NO, Ya existe' WHERE Folio_Nota = folio;
75 END;
76 /* LA CLAVE ES NULA */
77 DECLARE CONTINUE HANDLER FOR 1048
78 BEGIN
79     UPDATE carga_bruta SET Cargado = 'NO, Clave Nula' WHERE Folio_Nota is null;
80 END;
81 /* LA CLAVE CONTIENE UN CARACTER NO VALIDO */
```

The right sidebar shows a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."



The screenshot shows the continuation of the stored procedure 'Carga()'. It includes logic to handle data loading, including a loop to process multiple records and a cursor to iterate through the data.

```
81 /* LA CLAVE CONTIENE UN CARACTER NO VALIDO */
82 DECLARE CONTINUE HANDLER FOR 1205
83 BEGIN
84     UPDATE carga_bruta SET Cargado = 'NO, Clave No Val' WHERE Folio_Nota = folio;
85 END;
86 /* VIOLACION DE UN CHECK */
87 DECLARE CONTINUE HANDLER FOR 1062
88 BEGIN
89     UPDATE carga_bruta SET Cargado = 'NO, Act Check' WHERE Folio_Nota = folio;
90 END;
91
92 /* ----- VALOR DEL AUXILIAR PARA LA REPETICION ----- */
93 SELECT count(*) INTO auxiliar FROM carga_bruta;
94 /* ----- CURSOR ----- */
95 OPEN cursor_insert;
96 REPEAT
97     FETCH cursor_insert INTO folio, tel, ser, fecha, hora, est;
98
99     UPDATE carga_bruta SET Cargado = 'SI' WHERE Folio_Nota = folio;
100     INSERT INTO notas(Folio_Nota, Telefono_Cliente, Tipo_Servicio, Fecha_Entrega_Estimada, Fecha_Hora_Estimada, Estado)
101     VALUES (folio, tel, ser, fecha, hora, est);
102
103     SET auxiliar = auxiliar-1;
104 UNTIL auxiliar = 0 END REPEAT;
105 CLOSE cursor_insert;
106 END //
107 delimiter ;
108
```

The right sidebar shows a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

Se ejecuta el procedimiento.

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

```

104 UNTIL auxiliar = 0 END REPEAT;
105 CLOSE
106 END //
107 delimiter ;
108
109 /*----- ELIMINAR PROCEDIMIENTO ----- */
110 DROP procedure Carga;
111
112 /*----- LLEVAR A CABO EL PROCEDIMIENTO -----*/
113 Select * from notas;
114 select * from carga_bruta;
115 CALL Carga();
116
117 /* ELIMINACIÓN DE REGISTROS QUE SI SE INSERTARON */
118 DELETE FROM "tintoreria_aries"."notas"
119 WHERE ("Folio_Nota" = '825');
120

```

The Output window shows the following results:

#	Time	Action	Message	Duration / Fetch
554	08:18:26	DROP procedure Carga	Error Code: 1305. PROCEDURE tintoreria_aries.Carga does not exist	0.015 sec
555	08:19:17	CREATE PROCEDURE Carga() BEGIN DECLARE folio varchar(3); DECLARE tel v...	0 row(s) affected	0.000 sec
556	08:20:50	CALL Carga()	1 row(s) affected	0.109 sec

En la tabla de carga\_bruta se registra el por que no se insertaron los datos de esta tabla en la tabla de notas. Y carga con un sí los que se registraron de manera correcta.

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

```

113 Select * from notas;
114 select * from carga_bruta;
115 CALL Carga();
116
117 /* ELIMINACIÓN DE REGISTROS QUE SI SE INSERTARON */

```

The Output window shows the following results:

#	Time	Action	Message	Duration / Fetch
556	08:20:50	CALL Carga()	1 row(s) affected	0.109 sec
557	08:21:24	Select * from notas LIMIT 0, 1000	37 row(s) returned	0.000 sec / 0.000 sec
558	08:21:31	select * from carga_bruta LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

verificamos que en la tabla denotas de registraron de manera correcta los datos que tiene la tabla de carga\_bruta.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'practica' and 'tintoreria\_aries' expanded. The main editor shows a SQL script with three queries: a SELECT from 'notas', a SELECT from 'carga\_bruta', and a CALL to 'Carga()'. The 'Result Grid' shows the results of the first query, displaying columns: Folio\_Nota, Telefono\_Cliente, Tipo\_Servicio, Fecha\_Entrega\_Estimada, Fecha\_Hora\_Estimada, and Estatus. The 'Output' pane at the bottom shows the execution log with three entries, all successful.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

practica

tintoreria\_aries

Tables

Views

Stored Procedures

Functions

Administration Schemas

Information

No object selected

Object Info Session

Query Completed

Procedimiento

Limit to 1000 rows

113 • Select \* from notas;

114 • select \* from carga\_bruta;

115 • CALL Carga();

116

117 /\* ELIMINACIÓN DE REGISTROS QUE SI SE INSERTARON \*/

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content: 15

Folio_Nota	Telefono_Cliente	Tipo_Servicio	Fecha_Entrega_Estimada	Fecha_Hora_Estimada	Estatus
819	5586740704	Lavado	2021-11-05	19:30:00	En lavado
820	5586740704	Lavado en seco	2021-11-06	16:30:00	En Planta
825	5551901728	Planchado	2022-02-23	19:30:00	
826	5563802500	Planchado	2022-02-23	19:30:00	
827	5591098389	Lavado	2022-03-16	19:30:00	
828	5545303785	Reparacion	2021-02-16	19:30:00	
829	5596842072	Tefido	2021-02-17	19:30:00	
830	5504711160	Tefido	2022-04-06	19:30:00	
NULL	NULL	NULL	NULL	NULL	NULL

notes 113 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
557	08:21:24	Select * from notas LIMIT 0, 1000	37 row(s) returned	0.000 sec / 0.000 sec
558	08:21:31	select * from carga_bruta LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
559	08:23:45	Select * from notas LIMIT 0, 1000	37 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Form Editor

Field Types

Apply

Context Help

Snippets