

# Práctica 9: Diseño e implementación: Farmacias

## Administración y Diseño de Base de Datos

Akshay Chatani Chatani, Zuzanna Elzbieta Szalaty & Benjamin Paddags

A continuación se muestran algunas consultas realizadas:

SELECT \* FROM laboratorio:

MySQL Workbench interface showing the results of a query. The query is: `SELECT * FROM laboratorio`. The results are displayed in a table with columns: #, codigo, nombre, telefono, contacto\_nombre, contacto\_apellido, calle, numero, cp. The table contains 11 rows of data. The Action Output pane shows the execution details of the query, including the time taken and the number of rows affected.

#	codigo	nombre	telefono	contacto_nombre	contacto_apellido	calle	numero	cp
1	1234	A	566643455	GONZALO	MOHAMED	AVENIDA DE ESPAÑA	25	35315
2	2345	C	179686356	KHADILJA	BENITEZ	CARRERA MADRID	41	28891
3	3456	E	362732533	MARIA ELENA	CORRALES	RONDA REAL	81	48294
4	4321	B	756718387	MARIA CARMEN	CANO	CALLE IGLESIA	11	29632
5	4567	B	097840138	JUAN IGNACIO	RIVERO	CORREDERA IGLESIA	29	34976
6	5432	D	297718438	JUAN DIOS	GALAN	CORREDERA MAYOR	92	44174
7	5678	D	442122778	CARLOS MANUEL	VALDES	RONDA PEDRALBES	18	08527
8	6543	A	300347466	MARIA SOL	ORTEGA	VEREDA IGLESIA	30	43284
9	6789	F	424862282	ALEJANDRA	ROVIRA	VÍA HORNO	87	19828
10	7654	C	660829480	ESPERANZA	GARCIA	TRAVESIA PEDRALBES	25	46533
11	8765	E	442122779	SARAY	GARZON	CALLEJÓN PEDRALBES	33	22787

SELECT \* FROM medicamento :

MySQL Workbench interface showing the results of a query. The query is: `SELECT * FROM medicamento`. The results are displayed in a table with columns: #, codigo, nombre, tipo\_lgc, precio, receta, laboratorio.codigo, posologia. The table contains 10 rows of data. The Action Output pane shows the execution details of the query, including the time taken and the number of rows returned.

#	codigo	nombre	tipo_lgc	precio	receta	laboratorio.codigo	posologia
1	0123	ceftizima	6.5	12	0	2489	00:00:01
2	1234	ibuprofeno	6.5	12	1	1244	00:00:01
3	2345	ebastina	6.5	2	0	2345	00:00:06
4	3456	omeprazol	6.5	45	1	2462	00:00:04
5	4567	enalapril	6.5	5	0	4667	00:00:01
6	5678	lovastatina	6.5	10	1	8922	00:00:04
7	6789	fluoxetina	6.5	15	0	2375	00:00:01
8	7890	paroxetina	6.5	9	1	5382	00:00:01
9	8901	estradiol	6.5	8	0	9812	00:00:04
10	9012	diazepam	6.5	11	1	3587	00:00:06

```
SELECT * FROM compra_medicamento
```

The screenshot shows the MySQL Workbench interface. The query editor contains a series of INSERT statements into the 'compra' table of the 'mydb' database. The statements insert data for various medications, including 'cebridina', 'ibuprofeno', 'omeprazol', 'lovasitina', 'fluoxetina', 'paroxetina', 'estradiol', and 'diazepam'. The 'fecha\_hora' column is populated with dates from 2019-01-04 to 2019-01-11. The 'Action Output' pane at the bottom shows the execution of the query, indicating that 12 rows were returned and the operation was successful.

#	Importe_linea	unidades	lgic	compra.fecha_hora	medicamento.codigo
1	21	3	6.5	2019-01-02 00:00:00	2233
2	23	2	6.5	2019-01-01 00:00:00	1122
3	31	2	6.5	2019-01-12 00:00:00	1212
4	32	4	6.5	2019-01-09 00:00:00	1010
5	34	3	6.5	2019-01-08 00:00:00	8899
6	43	6	6.5	2019-01-05 00:00:00	5566
7	43	2	6.5	2019-01-07 00:00:00	7788
8	55	4	6.5	2019-01-03 00:00:00	3344
9	56	4	6.5	2019-01-10 00:00:00	1111
10	67	5	6.5	2019-01-04 00:00:00	4455
11	89	1	6.5	2019-01-11 00:00:00	1313
12	98	1	6.5	2019-01-06 00:00:00	6677

```
SELECT * FROM medicamento WHERE precio>5
```

The screenshot shows the MySQL Workbench interface. The query editor contains a SELECT statement that retrieves all medications with a price greater than 5. The 'Action Output' pane at the bottom shows the execution of the query, indicating that 8 rows were returned. The 'Result Grid' pane displays the results of the query, showing columns for 'codigo', 'nombre', 'tipo\_lgic', 'precio', 'receta', 'laboratorio.codigo', and 'posologia'.

#	codigo	nombre	tipo_lgic	precio	receta	laboratorio.codigo	posologia
1	0123	cebridina	6.5	12	0	2489	00:00:01
2	1234	ibuprofeno	6.5	12	1	1244	00:00:01
3	3456	omeprazol	6.5	45	1	2462	00:00:04
4	5678	lovasitina	6.5	10	1	8922	00:00:04
5	6789	fluoxetina	6.5	15	0	2375	00:00:06
6	7890	paroxetina	6.5	9	1	5382	00:00:01
7	8901	estradiol	6.5	8	0	9812	00:00:04
8	9012	diazepam	6.5	11	1	3587	00:00:06

```
SELECT telefono FROM laboratorio WHERE contacto_apellido='GALAN';
```

The screenshot shows the MySQL Workbench interface. The query editor contains a script with the following SQL statements:

```
312 * INSERT INTO `mydb`.`compra` (`importe`, `fechahora`) VALUES ('12', '2019-01-11');
313 * COMMIT;
314
315
316 DELIMITER $$
317
318 * DROP TRIGGER IF EXISTS prohibir_cambios$$
319 * CREATE TRIGGER prohibir_cambios
320 BEFORE UPDATE ON pago_credito FOR EACH ROW
321 BEGIN
322 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Prohibido cambiar el importe una vez esté pagado";
323 END$$
```

The 'Result Grid' shows the results of the query:

#	telefono
1	297718438

The 'Action Output' pane shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
126	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000083 sec
127	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000079 sec
128	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000077 sec
129	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000077 sec
130	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000076 sec
131	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000075 sec
132	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.00015 sec
133	19:25:15	COMMIT	0 row(s) affected	0.079 sec
134	19:25:15	DROP TRIGGER IF EXISTS prohibir_cambios	0 row(s) affected, 1 warning(s): 1360 Trigger does not exist	0.00012 sec
135	19:25:15	CREATE TRIGGER prohibir_cambios BEFORE UPDATE O...	0 row(s) affected	0.102 sec
136	19:25:15	SELECT telefono FROM laboratorio WHERE contacto_ap...	1 row(s) returned	0.00028 sec / 0.000...

```
SELECT precio FROM medicamento WHERE nombre='omeprazol';
```

The screenshot shows the MySQL Workbench interface. The query editor contains the same SQL script as the previous screenshot.

The 'Result Grid' shows the results of the query:

#	precio
1	45

The 'Action Output' pane shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
128	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000077 sec
129	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000077 sec
130	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000076 sec
131	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.000075 sec
132	19:25:15	INSERT INTO `mydb`.`compra` (`importe`, `fechahora`)	1 row(s) affected	0.00015 sec
133	19:25:15	COMMIT	0 row(s) affected	0.079 sec
134	19:25:15	DROP TRIGGER IF EXISTS prohibir_cambios	0 row(s) affected, 1 warning(s): 1360 Trigger does not exist	0.00012 sec
135	19:25:15	CREATE TRIGGER prohibir_cambios BEFORE UPDATE O...	0 row(s) affected	0.102 sec
136	19:25:15	SELECT telefono FROM laboratorio WHERE contacto_ap...	1 row(s) returned	0.00028 sec / 0.000...
137	19:26:38	SELECT precio FROM medicamento WHERE nombre='om...	Error Code: 1054. Unknown column 'omeprazol' ...	0.038 sec
138	19:26:50	SELECT precio FROM medicamento WHERE nombre='om...	1 row(s) returned	0.00044 sec / 0.000...