

## Overview

insert,delete,update

mysql constraints

sql data types(string,numeric,time and date)

**Insert,delete,update**

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • use hr;
2 • desc departments;
3 • insert into departments values(300,'HR',100,1200);
4 • select * from departments;
```

The result grid displays the following data:

department_id	department_name	manager_id	location_id
200	Operations	NULL	1700
210	IT Support	NULL	1700
220	NOC	NULL	1700
230	IT Helpdesk	NULL	1700
240	Government Sales	NULL	1700
250	Retail Sales	NULL	1700
260	Recruiting	NULL	1700
270	Payroll	NULL	1700
300	HR	100	1200
• NULL	NULL	NULL	NULL

The taskbar at the bottom shows the Windows Start button, a search bar, and various pinned icons.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
5
6 • update departments set manager_id=102 where department_id=300;
7 • select * from departments;
8
9
```

The result grid displays the following data:

department_id	department_name	manager_id	location_id
200	Operations	NULL	1700
210	IT Support	NULL	1700
220	NOC	NULL	1700
230	IT Helpdesk	NULL	1700
240	Government Sales	NULL	1700
250	Retail Sales	NULL	1700
260	Recruiting	NULL	1700
270	Payroll	NULL	1700
300	HR	102	1200
• NULL	NULL	NULL	NULL

The taskbar at the bottom shows the Windows Start button, a search bar, and various pinned icons.

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a browser window with tabs for 'Torry-Harris' and 'deleting row in mysql - Bing'. The main area displays a SQL query in 'Query 1' tab:

```
8  
9 • delete from departments where department_id=300;  
10 • select * from departments;  
11  
12
```

The results grid shows the following data:

department_id	department_name	manager_id	location_id
190	Contracting	NULL	1700
200	Operations	NULL	1700
210	IT Support	NULL	1700
220	NOC	NULL	1700
230	IT Helpdesk	NULL	1700
240	Government Sales	NULL	1700
250	Retail Sales	NULL	1700
260	Recruiting	NULL	1700
270	Payroll	NULL	1700
• NULL			

The status bar at the bottom indicates the date and time as 9/14/2021 7:45 PM.

## Inserting a row from one table to another

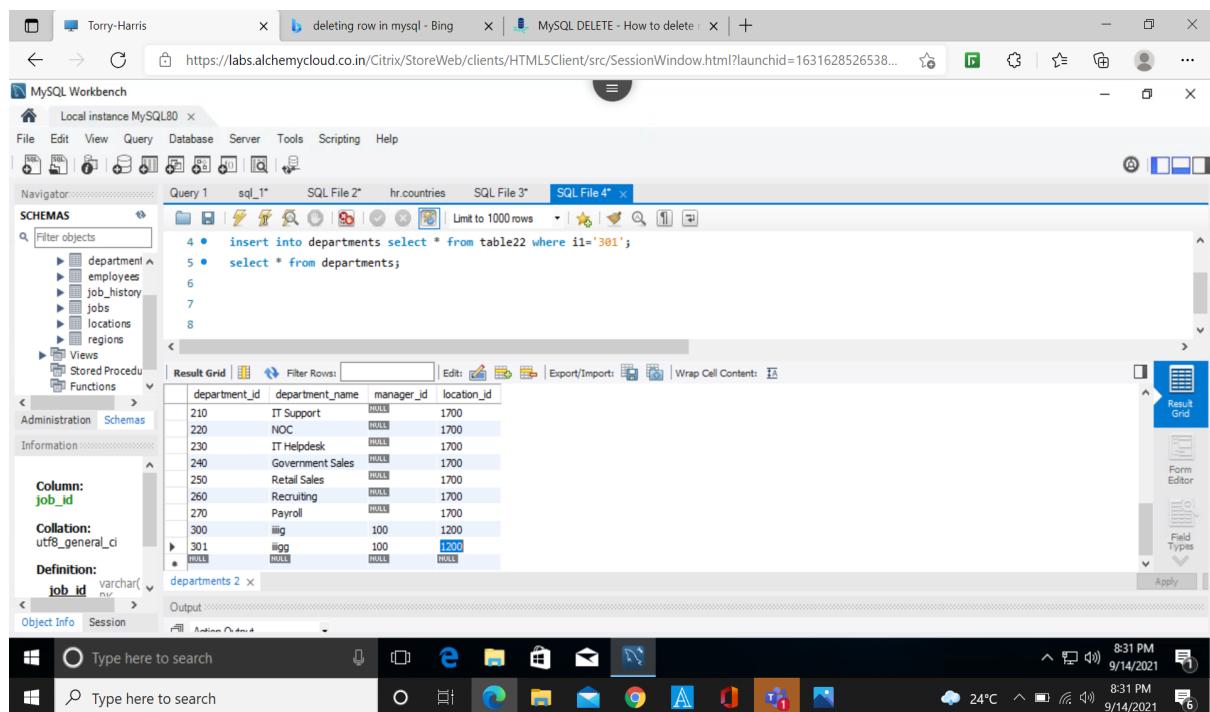
The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a browser window with tabs for 'Torry-Harris' and 'deleting row in mysql - Bing'. The main area displays a SQL query in 'Query 1' tab:

```
1 • use hr;  
2 • select * from table22;
```

The results grid shows the following data:

I1	I2	I3	I4
300	ilg	100	1200
301	ilgg	100	1200
302	ilggg	100	1200
303	lgggg	100	1200

The status bar at the bottom indicates the date and time as 9/14/2021 8:29 PM.



## TEAM 5 - SELECT, UPDATE and DELETE Commands

```
use hr;
desc departments;
```

	Field	Type	Null	Key	Default	Extra
▶	department_id	int unsigned	NO	PRI	NULL	
	department_name	varchar(30)	NO		NULL	
	manager_id	int unsigned	YES	MUL	NULL	
	location_id	int unsigned	YES	MUL	NULL	

```
select * from departments :
```

	department_id	department_name	manager_id	location_id
▶	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	40	Human Resources	203	2400
	50	Shipping	121	1500
	60	IT	103	1400
	70	Public Relations	204	2700
	80	Sales	145	2500
	90	Executive	100	1700
	100	Finance	108	1700
	110	Accounting	205	1700
	120	Treasury	NULL	1700
	130	Corporate Tax	NULL	1700
<hr/>				
	department_id	department_name	manager_id	location_id
	140	Control And Credit	NULL	1700
	150	Shareholder Servi...	NULL	1700
	160	Benefits	NULL	1700
	170	Manufacturing	NULL	1700
	180	Construction	NULL	1700
	190	Contracting	NULL	1700
	200	Operations	NULL	1700
	210	IT Support	NULL	1700
	220	NOC	NULL	1700
	230	IT Helpdesk	NULL	1700
	240	Government Sales	NULL	1700
	250	Retail Sales	NULL	1700
	260	Recruiting	NULL	1700
	270	Payroll	NULL	1700
*	NULL	NULL	NULL	NULL

### **#INSERTING A NEW RECORD( ROW ) TO THE TABLE :**

**insert into departments values(300, 'HR', 100, 1200);**

	department_id	department_name	manager_id	location_id
	260	Recruiting	NULL	1700
	270	Payroll	NULL	1700
	300	HR	100	1200
*	NULL	NULL	NULL	NULL

### **#UPDATING THE TABLE VALUES :**

**update departments set manager\_id=102 where department\_id=300;**

	department_id	department_name	manager_id	location_id
	260	Recruiting	NULL	1700
	270	Payroll	NULL	1700
*	300	HR	102	1200
*	NULL	NULL	NULL	NULL

### **#DELETING A RECORD FROM THE TABLE :**

**delete from departments where department\_id=300;**

	department_id	department_name	manager_id	location_id
	240	Government Sales	NULL	1700
	250	Retail Sales	NULL	1700
*	260	Recruiting	NULL	1700
*	270	Payroll	NULL	1700
*	NULL	NULL	NULL	NULL

### **#ADDING A NEW COLUMN TO THE TABLE :**

**alter table departments add(country\_name VARCHAR(5));**

	department_id	department_name	manager_id	location_id	country_name
▶	10	Administration	200	1700	NULL
	20	Marketing	201	1800	NULL
	30	Purchasing	114	1700	NULL
	40	Human Resources	203	2400	NULL
	50	Shipping	121	1500	NULL

### **#DELETING A COLUMN FROM THE TABLE :**

**alter table departments drop column country\_name;**

	department_id	department_name	manager_id	location_id
▶	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	40	Human Resources	203	2400
	50	Shipping	121	1500

# All Commands

```
use hr;  
desc departments;  
select * from departments;
```

```
#updating the table values  
update departments set manager_id=102 where department_id=300;
```

```
#inserting a new row to the table  
insert into departments values(300,'HR',100,1200,'IN');
```

```
#adding a new column to the table  
alter table departments add(country_name VARCHAR(5));
```

```
#deleting a row form the table  
delete from departments where department_id=300;
```

```
#deleting a column from a table  
alter table departments drop column country_name
```

## MYSQL CONSTRAINTS

### NOT NULL

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the following SQL code is run:

```
use persondb;  
create table student(  
    id int, last_name text not null, first_name text not null, city varchar(30));  
insert into student values(1,'rani','smriti','jharkhand');  
insert into student values(2,null,'mona','jaipur');  
desc student;
```

In the Output tab, the results of the actions are listed:

#	Action	Time	Message	Duration / Fetch
433	create table student(id int, last_name text not null, first_name text not null, city varchar(30))	19:55:50	0 rows(s) affected	0.437 sec
434	insert into student(1,'smriti','rani','jharkhand')	19:55:57	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'rani' at line 1	0.000 sec
435	insert into student(1,'rani','smriti','jharkhand')	19:56:33	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'smriti' at line 1	0.000 sec
436	insert into student values(1,'rani','smriti','jharkhand')	19:57:04	1 row(s) affected	0.063 sec
437	insert into student values(2,'mona','jaipur')	19:57:17	Error Code: 1136. Column count doesn't match value count at row 1	0.000 sec
438	insert into student values(2,null,'mona','jaipur')	19:57:47	Error Code: 1048. Column 'last_name' cannot be null	0.000 sec

The screenshot shows the MySQL Workbench interface. In the top tab bar, the active schema is 'hr-schema-mysql'. Below it, there are tabs for 'Administration - Server Status', 'SQL File 2\*', 'SQL File 4\*', 'new\_schema - Schema', 'persondb - Schema', and 'SQL File 5\*'. The main area displays the following SQL code:

```

3   id int, last_name text not null, first_name text not null, city varchar(30));
4
5 • insert into student values(1,'rani','smriti','jharkhand');
6 • insert into student values(2,null,'mona','jaipur');
7 • desc student;

```

Below the code, a 'Result Grid' shows the table structure:

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
last_name	text	NO		NULL	
first_name	text	NO		NULL	
city	varchar(30)	YES		NULL	

## UNIQUE

The screenshot shows the MySQL Workbench interface. In the top tab bar, the active schema is 'hr-schema-mysql'. Below it, there are tabs for 'Administration - Server Status', 'SQL File 2\*', 'SQL File 4\*', 'new\_schema - Schema', 'persondb - Schema', and 'SQL File 5\*'. The main area displays the following SQL code:

```

21
22
23 • create table hairstyles(
24   id int, name varchar(40) unique);
25
26 • insert hairstyles(id,name)values(1,'ponytail'),(2,'frenchbraid');
27
28 • insert hairstyles(id,name)values(1,'fishtailbraid'),(2,'frenchbraid');
29 • desc hairstyles;
30
31

```

Below the code, an 'Output' section shows the execution log:

#	Time	Action	Message	Duration /
457	20:31:29	drop table hairstyles	0 row(s) affected	0.110 sec
458	20:31:42	create table hairstyles(id int, name varchar(40) unique)	0 row(s) affected	0.203 sec
459	20:31:49	insert hairstyles(id, name)values(1, 'ponytail'), (2, 'frenchbraid')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.016 sec
460	20:32:04	insert hairstyles(id, name)values(1, 'fishtailbraid'), (2, 'frenchbraid')	Error Code: 1062. Duplicate entry 'frenchbraid' for key 'hairstyles.name'	0.000 sec

The status bar at the bottom right shows the date and time as 15-09-2023, 20:32:04.

yccloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1631714705129#launchurl=Resources

The screenshot shows the MySQL Workbench interface. The top menu bar includes Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The main workspace has tabs for 'hr-schema-mysql\*', 'Administration - Server Status', 'SQL File 2\*', 'SQL File 4\*', 'new\_schema - Schema', and 'persondb - Schema'. The 'SQL File 2\*' tab contains the following SQL code:

```
24     id int, name varchar(40) unique);
25
26 •   insert hairstyles(id,name)values(1,'ponytail'),(2,'frenchbraid');
27
28 •   insert hairstyles(id,name)values(1,'fishtailbraid'),(2,'frenchbraid');
29 •   desc hairstyles;
```

Below the code is a 'Result Grid' table showing the structure of the 'hairstyles' table:

	Field	Type	Null	Key	Default	Extra
▶	id	int	YES		NULL	
	name	varchar(40)	YES	UNI	NULL	

## CHECK

The screenshot shows the MySQL Workbench interface. The top menu bar includes Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The main workspace has tabs for 'hr-schema-mysql\*', 'Administration - Server Status', 'SQL File 2\*', 'SQL File 4\*', 'new\_schema - Schema', 'persondb - Schema', and 'SQL File 1'. The 'SQL File 2\*' tab contains the following SQL code:

```
1 •   use persondb;
2 •   CREATE TABLE Persons(
3     ID int NOT NULL,
4     Name varchar(45) NOT NULL,
5     Age int CHECK(Age>=18));
6
7 •   INSERT INTO Persons(Id, Name, Age)
8     VALUES(1, 'smriti', 22),(2, 'hema', 35),(3, 'amit', 40);
9
10 •  INSERT INTO Persons(Id, Name, Age)
11    VALUES(1, 'rahul', 13);
12
13 •  select*from Persons;
```

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

	ID	Name	Age
▶	1	smriti	22
	2	hema	35
	3	amit	40

M\_SH Persons 3

to search

DEFAULT

hr-schema-mysql\* Administration - Server Status SQL File 2\* SQL File 4\* new\_schema - Schema persondb - Schema SQL File 5\*

```

1 • use persondb;
2 • CREATE TABLE places(
3     ID int NOT NULL,
4     Name varchar(30) NOT NULL,
5     Age int,
6     City varchar(20) DEFAULT 'Mumbai');
7
8 • INSERT INTO places(Id, Name, Age, City)
9     VALUES(1, 'shristi', 22, 'jharkhand'),(2, 'neha', 21, 'Delhi'),(3, 'Sana', 35, 'kolkata');
10
11 • INSERT INTO places(Id, Name, Age)
12     VALUES(1, 'sneha', 36);
13
14 • select * from places;

```

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

	ID	Name	Age	City
▶	1	shristi	22	jharkhand
	2	neha	21	Delhi
	3	Sana	35	kolkata
	1	sneha	36	Mumbai

SH

PRIMARY KEY

Database Server Tools Scripting Help

hr-schema-mysql\* Administration - Server Status SQL File 2\* SQL File 4\* new\_schema - Schema persondb - Schema SQL File 5\*

```
10 • create table Persons(
11     id int not null primary key,
12     name varchar(45)not null,
13     age int,
14     city varchar(30));
15 • insert into Persons(id,name,age,city)
16     values(1, 'smriti', 22,'jharkhand'),(2, 'hema', 35,'kolkata'),(3, 'amit' ,40,'delhi');
17
18 • insert into Persons(id, name,age,city)
19     values(1, 'rahul', 13,'mumbai');
20
21 • select*from Persons;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
452	20:19:24	insert into Persons(id,name,age,city) values(1, 'smriti', 22,jharkhand),(2, 'hema', 35,...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.031 sec
453	20:19:31	insert into Persons(id, name,age,city) values(1, 'rahul', 13,mumbai)	Error Code: 1062. Duplicate entry '1' for key 'persons.PRIMARY'	0.000 sec

search 8:25 PM 9/15/2021 20:25 23°C Light rain ENG 15-09-2021

search

306 x Database Server Tools Scripting Help

hr-schema-mysql\* Administration - Server Status SQL File 2\* SQL File 4\* new\_schema - Schema persondb - S

```
17
18 • insert into Persons(id, name,age,city)
19     values(1, 'rahul', 13,'mumbai');
20
21 • select*from Persons;
22
23
24
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	name	age	city
▶	1	smriti	22	jharkhand
▶	2	hema	35	kolkata
▶	3	amit	40	delhi
*	NULL	NULL	NULL	NULL

Persons 7 x

## SQL PRIMARY KEY on ALTER TABLE

The screenshot shows a MySQL Workbench interface with several tabs open. The main query editor contains the following SQL code:

```
11 • create table hairstyles(
12     id int, name varchar(40));
13
14 • insert hairstyles(id,name)values(1,'ponytail'),(2,'frenchbraid');
15 • alter table hairstyles add primary key(id);
16 • insert hairstyles(id,name)values(1,'fishtail'),(2,'frenchbraid');
17 • desc hairstyles;
```

The 'Output' pane displays the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
486	21:36:51	alter table hairstyles add primary key(id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.297 sec
487	21:37:05	insert hairstyles(id,name)values(1,'ponytail'),(2,'frenchbraid')	Error Code: 1062. Duplicate entry '1' for key 'hairstyles.PRIMARY'	0.000 sec
488	21:37:26	drop table hairstyles	0 row(s) affected	0.078 sec
489	21:37:34	create table hairstyles(id int, name varchar(40))	0 row(s) affected	0.140 sec
490	21:37:39	insert hairstyles(id,name)values(1,'ponytail'),(2,'frenchbraid')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.031 sec
491	21:37:49	alter table hairstyles add primary key(id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.421 sec
492	21:38:16	insert hairstyles(id,name)values(1,'fishtail'),(2,'frenchbraid')	Error Code: 1062. Duplicate entry '1' for key 'hairstyles.PRIMARY'	0.000 sec

The system tray at the bottom right shows the date and time as 9/15/2021 9:39 PM, and the weather as 23°C Light rain.

The screenshot shows a MySQL Workbench interface. In the top navigation bar, there are tabs for 'Outlook', 'Torry-Harris', and 'MySQL Constraints - javatpoint'. Below the tabs, the URL is displayed as [abs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1631721666104#launchurl=Resources%2FLaunchica%2FQkxS...](http://abs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1631721666104#launchurl=Resources%2FLaunchica%2FQkxS...). The main window displays a SQL editor with the following code:

```
13
14 •    insert hairstyles(id,name)values(1,'ponytail'),(2,'frenchraid');
15 •    alter table hairstyles add primary key(id);
16 •    insert hairstyles(id,name)values(1,'fishtail'),(2,'frenchbraid');
17 •    desc hairstyles;
```

Below the code, there is a 'Result Grid' table showing the structure of the 'hairstyles' table:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(40)	YES		NULL	

**To drop a PRIMARY KEY constraint, use the following SQL:**

The screenshot shows a MySQL Workbench interface. The main window displays a SQL editor with the following code:

```
15 • alter table hairstyles add primary key(id);
16 • insert hairstyles(id,name)values(1,'fishtail'),(2,'Frenchbraid');
17
18 • alter table hairstyles drop primary key;
19 • desc hairstyles;
```

Below the code, there is a table named 'Result Grid' showing the structure of the 'hairstyles' table:

Field	Type	Null	Key	Default	Extra
id	int	NO		HULL	
name	varchar(40)	YES		HULL	

The status bar at the bottom right indicates the following information: 9:44 PM, 9/15/2021, 21:43, 23°C Light rain, ENG, 15-09-2021.

## AUTO\_INCREMENT



1:3306 x

Query Database Server Tools Scripting Help

hr-schema-mysql\* Administration - Server Status SQL File 2\* SQL File 4\* new\_schema - Schema persondb - Schema SQL File 5\*

```

27
28 • insert into hairstyles(name)values('ponytail'),('frenchbraid'),('fishtailbraid'),('halfponytail');
29
30 • desc hairstyles;
31
32
33

```

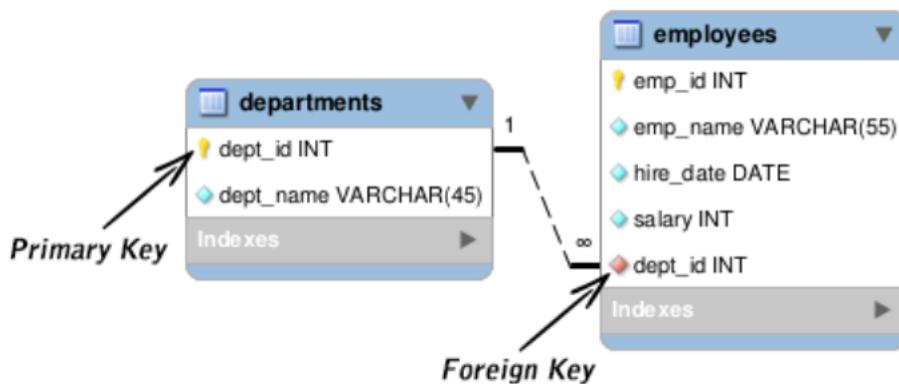
Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	<u>HULL</u>	auto_increment
name	varchar(40)	NO		<u>HULL</u>	

M\_SH Result 9 x

to search 23°C Light rain ENG

## FOREIGN KEY



Torry-Harris MySQL Constraints - javatpoint hemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1631714705129#launchurl=Resources%2FLaunchica%2FQkxS... ☆ \*

Database Server Tools Scripting Help

hr-schema-mysql\* Administration - Server Status SQL File 2\* SQL File 4\* new\_schema - Schema persondb - Schema SQL File 5\*

```
8
9
10 • create table persons(
11     person_id int not null primary key, name varchar(45)not null, age int, city varchar(30));
12
13 • insert into persons(person_id,name,age,city)
14     values(1, 'smriti', 22,'jharkhand'),(2, 'hema', 35,'kolkata'),(3, 'amit' ,40,'delhi');
15
16 • select*from persons;
17
18 • create table orders(
19     order_id int not null primary key, order_num int not null, person_id int,
20     foreign key(person_id)references persons(person_id));
21
22 • insert into orders(order_id,order_num,person_id)values(1,500,2),(2,501,1),(3,502,3),(4,503,2);
23
24 • select*from orders;
25
26
```

Search 9:01 9/15 21:00 23°C Light rain ENG 15-09-2023

ni - Outlook    Torry-Harris    MySQL Constraints - javatpoint

labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1631714705129#launchurl=Resources%2FLaunchId

0.1.3306

Query Database Server Tools Scripting Help

Procedures    Schemas    localhost    Session

hr-schema-mysql\*    Administration - Server Status    SQL File 2\*    SQL File 4\*    new\_schema - Schema    persondb - Schema    SQL File 5\*

```
10 • create table persons(
11     person_id int not null primary key, name varchar(45)not null, age int, city varchar(30));
12
13 • insert into persons(person_id,name,age,city)
14     values(1, 'smriti', 22,'jharkhand'),(2, 'hema', 35,'kolkata'),(3, 'amit' ,40,'delhi');
15
16 • select*from persons;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

person_id	name	age	city
1	smriti	22	jharkhand
2	hema	35	kolkata
3	amit	40	delhi
HULL	HULL	HULL	HULL

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the tabs, there are several tool icons. The main workspace displays SQL code in a script editor:

```
18 • create table orders(
19     order_id int not null primary key, order_num int not null, person_id int,
20     foreign key(person_id)references persons(person_id);
21
22 •   insert into orders(order_id,order_num,person_id)values(1,500,2),(2,501,1),(3,502,3),(4,503,2);
23
24 •   select*from orders;
```

Below the code, a 'Result Grid' shows the data inserted into the 'orders' table:

order_id	order_num	person_id
1	500	2
2	501	1
3	502	3
4	503	2
• NULL	NULL	NULL

The status bar at the bottom right shows the date and time: 9:03 PM 9/15/2021, along with system information: 23°C Light rain, 21:03, ENG, 15-09-2021.

## SQL FOREIGN KEY on ALTER TABLE

To create a FOREIGN KEY constraint on the "PersonID" column when the "Orders" table is already created, use the following SQL:

ALTER TABLE Orders

ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

To allow naming of a FOREIGN KEY constraint, and for defining a FOREIGN KEY constraint on multiple columns, use the following SQL syntax:

ALTER TABLE Orders

MySQL:

ALTER TABLE Orders

DROP FOREIGN KEY FK\_PersonOrder;

```
ALTER TABLE Orders
```

```
DROP CONSTRAINT FK_PersonOrder;
```

## SQL Data Types

Data types are used to represent the nature of the data that can be stored in the database table. For example, in a particular column of a table, if we want to store a string type of data then we will have to declare a string data type of this column.

Data types mainly classified into three categories for every database.

String Data types

Numeric Data types

Date and time Data types

<b>CHAR(Size)</b>	It is used to specify a fixed length string that can contain numbers, letters, and special characters. Its size can be 0 to 255 characters. Default is 1.
<b>VARCHAR(Size)</b>	It is used to specify a variable length string that can contain numbers, letters, and special characters. Its size can be from 0 to 65535 characters.
<b>BINARY(Size)</b>	It is equal to CHAR() but stores binary byte strings. Its size parameter specifies the column length in the bytes. Default is 1.
<b>VARBINARY(Si ze)</b>	It is equal to VARCHAR() but stores binary byte strings. Its size parameter specifies the maximum column length in bytes.
<b>TEXT(Size)</b>	It holds a string that can contain a maximum length of 255 characters.
<b>TINYTEXT</b>	It holds a string with a maximum length of 255 characters.
<b>MEDIUMTEXT</b>	It holds a string with a maximum length of 16,777,215.
<b>LONGTEXT</b>	It holds a string with a maximum length of 4,294,967,295 characters.

<b>BLOB(size)</b>	It is used for BLOBS (Binary Large Objects). It can hold up to 65,535 bytes.
-------------------	------------------------------------------------------------------------------

## MySQL Numeric Data Types

<b>BIT(Size)</b>	It is used for a bit-value type. The number of bits per value is specified in size. Its size can be 1 to 64. The default value is 1.
<b>INT(size)</b>	It is used for the integer value. Its signed range varies from -2147483648 to 2147483647 and unsigned range varies from 0 to 4294967295. The size parameter specifies the max display width that is 255.
<b>INTEGER (size)</b>	It is equal to INT(size).
<b>FLOAT(size, d)</b>	It is used to specify a floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal point is specified by <b>d</b> parameter.
<b>FLOAT(p)</b>	It is used to specify a floating point number. MySQL used p parameter to determine whether to use FLOAT or DOUBLE. If p is between 0 to 24, the data type becomes FLOAT(). If p is from 25 to 53, the data type becomes DOUBLE().
<b>DOUBLE(size, d)</b>	It is a normal size floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal is specified by d parameter.

<b>DECIMAL (size, d)</b>	It is used to specify a fixed point number. Its size parameter specifies the total number of digits. The number of digits after the decimal parameter is specified by <b>d</b> parameter. The maximum value for the size is 65, and the default value is 10. The maximum value for <b>d</b> is 30, and the default value is 0.
<b>DEC(size, d)</b>	It is equal to DECIMAL(size, d).
<b>BOOL</b>	It is used to specify Boolean values true and false. Zero is considered as false, and nonzero values are considered as true.

## MySQL Date and Time Data Types

<b>DATE</b>	It is used to specify date format YYYY-MM-DD. Its supported range is from '1000-01-01' to '9999-12-31'.
<b>DATETIME (fsp)</b>	It is used to specify date and time combination. Its format is YYYY-MM-DD hh:mm:ss. Its supported range is from '1000-01-01 00:00:00' to 9999-12-31 23:59:59'.
<b>TIMESTA MP(fsp)</b>	It is used to specify the timestamp. Its value is stored as the number of seconds since the Unix epoch('1970-01-01 00:00:00' UTC). Its format is YYYY-MM-DD hh:mm:ss. Its supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC.
<b>TIME(fsp)</b>	It is used to specify the time format. Its format is hh:mm:ss. Its supported range is from '-838:59:59' to '838:59:59'
<b>YEAR</b>	It is used to specify a year in four-digit format. Values allowed in four digit format from 1901 to 2155, and 0000.

## SQL Server Data Types

### SQL Server String Data Type

<b>char(n)</b>	It is a fixed width character string data type. Its size can be up to
----------------	-----------------------------------------------------------------------

	8000 characters.
<b>varchar(n)</b>	It is a variable width character string data type. Its size can be up to 8000 characters.
<b>varchar(max)</b>	It is a variable width character string data types. Its size can be up to 1,073,741,824 characters.
<b>text</b>	It is a variable width character string data type. Its size can be up to 2GB of text data.
<b>nchar</b>	It is a fixed width Unicode string data type. Its size can be up to 4000 characters.
<b>nvarchar</b>	It is a variable width Unicode string data type. Its size can be up to 4000 characters.
<b>ntext</b>	It is a variable width Unicode string data type. Its size can be up to 2GB of text data.
<b>binary(n)</b>	It is a fixed width Binary string data type. Its size can be up to 8000 bytes.
<b>varbinary</b>	It is a variable width Binary string data type. Its size can be up to 8000 bytes.
<b>image</b>	It is also a variable width Binary string data type. Its size can be up to 2GB.

## SQL Server Numeric Data Types

<b>bit</b>	It is an integer that can be 0, 1 or null.
------------	--------------------------------------------

<b>tinyint</b>	It allows whole numbers from 0 to 255.
<b>Smallint</b>	It allows whole numbers between -32,768 and 32,767.
<b>Int</b>	It allows whole numbers between -2,147,483,648 and 2,147,483,647.
<b>bigint</b>	It allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807.
<b>float(n)</b>	It is used to specify floating precision number data from -1.79E+308 to 1.79E+308. The n parameter indicates whether the field should hold the 4 or 8 bytes. Default value of n is 53.
<b>real</b>	It is a floating precision number data from -3.40E+38 to 3.40E+38.
<b>money</b>	It is used to specify monetary data from -922,337,233,685,477.5808 to 922,337,203,685,477.5807.

### SQL Server Date and Time Data Type

<b>datetime</b>	It is used to specify date and time combination. It supports range from January 1, 1753, to December 31, 9999 with an accuracy of 3.33 milliseconds.
<b>datetime2</b>	It is used to specify date and time combination. It supports range from January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds
<b>date</b>	It is used to store date only. It supports range from January 1, 0001 to December 31, 9999
<b>time</b>	It stores time only to an accuracy of 100 nanoseconds
<b>timestamp</b>	It stores a unique number when a new row gets created or modified. The time stamp value is based upon an internal clock and does not correspond to real time. Each table

may contain only one-time stamp variable.

## SQL Server Other Data Types

<b>Sql_variant</b>	It is used for various data types except for text, timestamp, and ntext. It stores up to 8000 bytes of data.
<b>XML</b>	It stores XML formatted data. Maximum 2GB.
<b>cursor</b>	It stores a reference to a cursor used for database operations.