MySQL INSERT Statement

MySQL INSERT statement is used to store or add data in MySQL table within the database. We can perform insertion of records in two ways using a single query in MySQL:

```
    Insert record in a single row

    Insert record in multiple rows
```

```
CREATE TABLE People(
    id int NOT NULL AUTO_INCREMENT,
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    age int,
    PRIMARY KEY (id)
);
```

Insert record in a single row

```
INSERT INTO People (id, name, occupation, age)
VALUES (101, 'Peter', 'Engineer', 32);
```

```
SELECT * FROM People;
```

Insert record in a Multiple row

```
INSERT INTO People VALUES
(102, 'Joseph', 'Developer', 30),
(103, 'Mike', 'Leader', 28),
```

```
(104, 'Stephen', 'Scientist', 45);
```

```
SELECT * FROM People;
```

Following Constraints

```
INSERT INTO People (name, occupation)
VALUES ('Stephen', 'Scientist'), ('Bob', 'Actor');
```

```
SELECT * FROM People;
```

Example

```
CREATE TABLE trainer(
    id int NOT NULL AUTO_INCREMENT,
    Trainer varchar(45) NOT NULL,
    Course_Name varchar(35) NOT NULL,
    Email varchar(55) NOT NULL,
    PRIMARY KEY (id)
);
```

```
INSERT INTO trainer (id, Trainer, Course_Name, Email)
VALUES (1,'Peter', 'Ds', 'petor@learnai.co.in'),
(2,'Mike', 'Java', 'mike@learnai.co.in'),
(3,'James', 'Python', 'james@learnai.co.in'),
(4,'Robin', 'Android', 'robin@learnai.co.in');
```

```
UPDATE trainer
SET email = 'peter@learnai.co.in'
WHERE course_name = 'Ds';
```

```
UPDATE trainer
SET Trainer = 'Marry', course_name = 'Content Writer', Email='marry@gmail.com'
```

```
WHERE id = 2;
```

```
UPDATE trainer
SET email = REPLACE(email,'@gmail.com','@learnai.co.in')
WHERE Course_Name = 'Content Writer';
```

```
DELETE FROM trainer WHERE id=3;
```

Select

```
CREATE TABLE Employee_detail (
    id INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(35),
    Email VARCHAR(35),
    Phone bigint,
    City VARCHAR(35),
    Working_hours int);

INSERT INTO Employee_detail(id, Name, Email, Phone, City, Working_hours)
VALUES (1,'Peter', 'peter@gmail.com',1234567890,'TEXAS',12),
(2,'Suzi', 'Suzi@gmail.com',1234567890,'California',10),
(3,'Joseph', 'Joseph@gmail.com',1234587890,'NJ',14),
(4,'Alex', 'Alex@gmail.com',9234567890,'LA',9),
(5,'Mark', 'Mark@gmail.com',3234567890,'Washington',12),
(6,'Stephen', 'Stephen@gmail.com',1234567690,'New York',10);

SELECT Name FROM employee_detail;

SELECT Name, Email, City FROM employee_detail;

SELECT * FROM employee_detail;

REPLACE INTO employee_detail (id, City)
VALUES(4,'Amsterdam');

SELECT * FROM employee_detail;

REPLACE INTO employee_detail
```

```
SET ID = 1,
    Name = 'Mike',
    Email = 'mike@gmail.com' ,
    Phone = 1546854556,
    City = 'NJ',
    Working_hours = 15;

SELECT * FROM employee_detail;
```

```
SELECT *
FROM employee_detail
WHERE City = 'NJ';
```

```
SELECT *
FROM employee_detail
WHERE City = 'NJ'
AND Working_hours < 15;
```

```
SELECT *
FROM employee_detail
WHERE City = 'NJ'
OR City = 'Washington';
```

```
SELECT *
FROM employee_detail
WHERE (City = 'NJ' AND Name = 'Mike')
OR (Working_hours < 15);
```

```
SELECT DISTINCT City
FROM employee_detail;
```

```
SELECT DISTINCT name, city
FROM employee_detail;
```

```
SELECT *
FROM employee_detail
```

```
WHERE id <= 3;
```

```
CREATE TABLE officers(
    officer_id int NOT NULL,
    officer_name varchar(45) NOT NULL,
    address varchar(35) NOT NULL);

INSERT INTO officers (officer_id, officer_name, address)
VALUES (1,'Ajeet', 'Mau'),
(2,'Deepika', 'Lucknow'),
(3,'Vimal', 'Faizabad'),
(4,'Rahul', 'Lucknow');

select * from officers;
```

In [ ]:
```
SELECT DISTINCT address
FROM officers;
```

In [ ]:
```
SELECT *
FROM officers
WHERE address = 'Lucknow';
```

```
CREATE TABLE students(
    student_id int NOT NULL,
    student_name varchar(45) NOT NULL,
    course_name varchar(35) NOT NULL);

INSERT INTO students (student_id, student_name, course_name)
VALUES (1,'Aryan', 'Java'),
(2,'Rohini', 'Hadoop'),
(3,'Lallu', 'MongoDB'),
(4,'Ravi', 'Python');

select * from students;
```

```
SELECT officers.officer_id, students.student_name
FROM students
```

```
INNER JOIN officers
ON students.student_id = officers.officer_id;
```

```
SELECT officers.officer_id, students.student_name
FROM officers
LEFT OUTER JOIN students
ON officers.officer_id = students.student_id;
```

```
SELECT *
FROM officers
WHERE address = 'Lucknow'
ORDER BY officer_name;
```

In [ ]:
```
SELECT *
FROM officers
WHERE address = 'Lucknow'
ORDER BY officer_name ASC;
```

In [ ]:
```
SELECT *
FROM officers
WHERE address = 'Lucknow'
ORDER BY officer_name DESC;
```

In [ ]:
```
SELECT officer_name, address
FROM officers
WHERE officer_id < 5
ORDER BY officer_name DESC, address ASC;
```

In [ ]:
```
SELECT address, COUNT(*)
FROM officers
GROUP BY address;
```

```
CREATE TABLE employees(
    emp_id int NOT NULL,
    emp_name varchar(45) NOT NULL,
    working_date DATE NOT NULL,
    working_hours int NOT NULL);
```

```
INSERT INTO employees (emp_id, emp_name, working_date,working_hours)
VALUES (1,'Ajeet', '2015-01-24', 12),
(2,'Ayan', '2015-01-24', 10),
(3,'Milan', '2015-01-24', 9),
(4,'Ruchi', '2015-01-24', 6),
(1,'Ajeet', '2015-01-25', 12),
(2,'Ayan', '2015-01-25', 10),
(3,'Milan', '2015-01-25', 6),
(4,'Ruchi', '2015-01-25', 9),
(1,'Ajeet', '2015-01-26', 12),
(3,'Milan', '2015-01-26', 9);

select * from employees;
```

In [ ]:
```
SELECT emp_name, SUM(working_hours) AS "Total working hours"
FROM employees
GROUP BY emp_name;
```

In [ ]:
```
SELECT emp_name, MIN(working_hours) AS "Minimum working hour"
FROM employees
GROUP BY emp_name;
```

In [ ]:
```
SELECT emp_name, MAX(working_hours) AS "Maximum working hour"
FROM employees
GROUP BY emp_name;
```

In [ ]:
```
SELECT emp_name, AVG(working_hours) AS "Average working hour"
FROM employees
GROUP BY emp_name;
```

In [ ]:
```
SELECT emp_name, SUM(working_hours) AS "Total working hours"
FROM employees
GROUP BY emp_name
HAVING SUM(working_hours) > 16;
```

In [ ]:
```sql
SELECT emp_name, SUM(working_hours) AS "Total working hours"
FROM employees
GROUP BY emp_name
HAVING SUM(working_hours) > 25;
```

In [ ]:
```sql
SELECT emp_name, SUM(working_hours) AS "Total working hours"
FROM employees
GROUP BY emp_name
HAVING SUM(working_hours) > 16;
```

In [ ]:
```sql
SELECT *
FROM employees
LIMIT 5;
```

In [ ]:
```sql
SELECT *
FROM employees
ORDER BY emp_name DESC
LIMIT 2;
```