

## Queries to Modify and Test your Database

**INSERT** - To insert data into a new row into an existing table.

### With Good Data:

#### *SQL Query:*

```
INSERT INTO activetime
(ActiveTimeId,VeryActiveMinutes,FairlyActiveMinutes,LightlyActiveMinutes,
SedentaryMinutes)
VALUES (500, 27, 30, 50, 50);
```

#### *Results:*

```
8 • INSERT INTO activetime (ActiveTimeId,VeryActiveMinutes,FairlyActiveMinutes,LightlyActiveMinutes, SedentaryMinutes)
9   VALUES (500, 27, 30, 50, 50);
10 • SELECT * FROM activetime WHERE ActiveTimeId = 500;
```

11

ActiveTimeId	VeryActiveMinutes	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes
500	27	30	50	50
NULL	NULL	NULL	NULL	NULL

activetime 9 x

Output

#	Time	Action	Message
1	18:27:54	INSERT INTO activetime (ActiveTimeId,VeryActiveMinutes,FairlyActiveMinutes,LightlyActiveMinutes, SedentaryMinutes) VALUES (500, 27, 30...	1 row(s) affected
2	18:27:54	SELECT * FROM activetime WHERE ActiveTimeId = 500 LIMIT 0, 400	1 row(s) returned

### With Bad Data: -- Error (Trying to insert duplicate record with same Primary Key)

#### *SQL Query:*

```
INSERT INTO activetime (ActiveTimeId)
VALUES (500);
```

#### *Results:*

```

12 -- Error (Trying to insert duplicate record with same Primary Key)
13 • INSERT INTO activetime (ActiveTimeId)
14   VALUES (500);
15

```

Output			Message
#	Time	Action	
1	18:33:49	INSERT INTO activetime (ActiveTimeId) VALUES (500)	Error Code: 1062. Duplicate entry '500' for key 'activetime.PRIMARY'

## UPDATE - To Update data in an existing row in a table.

With Good Data:

### SQL query:

```

UPDATE activetime
SET VeryActiveMinutes = 34
WHERE ActiveTimeId = 500;

```

```

SELECT * FROM activetime WHERE ActiveTimeId = 500;

```

### Results:

```

20 -- Success
21 • UPDATE activetime
22   SET VeryActiveMinutes = 34
23   WHERE ActiveTimeId = 500;
24
25 • SELECT * FROM activetime WHERE ActiveTimeId = 500;
26
27 -- Error (Trying to update string value to a integer datatype field)

```

Result Grid					
Filter Rows:					
	ActiveTimeId	VeryActiveMinutes	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes
▶	500	34	30	50	50
*	NULL	NULL	NULL	NULL	NULL

activetime 13 x

Output			Message
#	Time	Action	
1	18:39:38	UPDATE activetime SET VeryActiveMinutes = 34 WHERE ActiveTimeId = 500	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2	18:39:38	SELECT * FROM activetime WHERE ActiveTimeId = 500 LIMIT 0, 400	1 row(s) returned

## With Bad Data: Trying to update string value to a integer datatype field

### *SQL query:*

```
UPDATE activetime
SET VeryActiveMinutes = 'Test'
WHERE ActiveTimeId = 500;
```

### *Results:*

```
27 -- Error (Trying to update string value to a integer datatype field)
28 • UPDATE activetime
29   SET VeryActiveMinutes = 'Test'
30   WHERE ActiveTimeId = 500;
31
```

Output

#	Time	Action	Message
1	18:42:50	UPDATE activetime SET VeryActiveMinutes = 'Test' WHERE ActiveTimeId = 500	Error Code: 1366. Incorrect integer value: 'Test' for column 'VeryActiveMinutes' at row 1

## DELETE – delete data a row in an existing table

### With Good Data:

### *SQL query:*

```
DELETE FROM activeTime
WHERE ActiveTimeId = 500;
```

### *Results:*

```
>>
34 -- Success
35 • DELETE FROM activeTime
36   WHERE ActiveTimeId = 500;
37
```

Output

#	Time	Action	Message
1	18:45:33	DELETE FROM activeTime WHERE ActiveTimeId = 500	1 row(s) affected

## With Bad Data: DataType mismatch

### *SQL query:*

```
DELETE FROM activeTime
WHERE ActiveTimeId = 'Test';
```

### *Results:*

```

38 -- Error (DataType mismatch)
39 • DELETE FROM activeTime
40 WHERE ActiveTimeId = 'Test';
41
42

```

Output			
Action Output			
#	Time	Action	Message
1	18:48:59	DELETE FROM activeTime WHERE ActiveTimeId = 'Test'	Error Code: 1292. Truncated incorrect DOUBLE value: 'Test'

**Function: GET\_AVG\_SLEEPTIME (To get the avg sleep time of user, Input Parameter: User Id)**

**With Good data –**

***SQL code:***

USE fitbit\_db;

DROP FUNCTION IF EXISTS get\_avg\_sleeptime;

DELIMITER //

CREATE FUNCTION get\_avg\_sleeptime (  
user\_id BIGINT)

RETURNS double

DETERMINISTIC READS SQL DATA

BEGIN

DECLARE avg\_sleeptime double;

SELECT

ROUND(AVG(TotalMinutesAsleep), 2)

INTO avg\_sleeptime FROM

sleeptime

JOIN

dailyactivity ON sleeptime.SleepId = dailyactivity.SleepId

WHERE

Id = user\_id;

RETURN round((avg\_sleeptime / 60.0), 2);

END//

select get\_avg\_sleeptime(1503960366);

***Results:***

Limit to 400 rows

```

3 • DROP FUNCTION IF EXISTS get_avg sleeptime;
4
5 DELIMITER //
6 • CREATE FUNCTION get_avg sleeptime (
7   user_id BIGINT)
8   RETURNS double
9   DETERMINISTIC READS SQL DATA
10 BEGIN
11   DECLARE avg_sleeptime double;
12   SELECT
13     ROUND(AVG(TotalMinutesAsleep), 2)
14   INTO avg_sleeptime FROM
15     sleeptime
16     JOIN
17     dailyactivity ON sleeptime.SleepId = dailyactivity.SleepId
18   WHERE
19     Id = user_id;
20   RETURN round((avg_sleeptime / 60.0), 2);
21 END//
22
23 • select get_avg sleeptime(1503960366);

```

Result Grid

	get_avg sleeptime(1503960366)
▶	6.21

Result 1 ×

Output

Action Output

#	Time	Action	Message
✓ 1	00:44:11	USE fitbit_db	0 row(s) affected
✓ 2	00:44:11	DROP FUNCTION IF EXISTS get_avg sleeptime	0 row(s) affected
✓ 3	00:44:11	CREATE FUNCTION get_avg sleeptime ( user_id BIGINT...	0 row(s) affected
✓ 4	00:44:11	select get_avg sleeptime(1503960366) LIMIT 0, 400 ;	1 row(s) returned

**With Bad data : Passing wrong data type.****SQL code:**

```
select get_avg_sleeptime("user A");
```

**Results:**

```
22
23 • select get_avg_sleeptime("user A");
```

Output				
Action Output				
#	Time	Action	Message	
✓ 1	00:53:21	USE fitbit_db	0 row(s) affected	
✓ 2	00:53:21	DROP FUNCTION IF EXISTS get_avg_sleeptime	0 row(s) affected	
✓ 3	00:53:21	CREATE FUNCTION get_avg_sleeptime ( user_id BIGINT)...	0 row(s) affected	
✗ 4	00:53:21	select get_avg_sleeptime("user A") LIMIT 0, 400 ;	Error Code: 1366. Incorrect integer value: 'user A' for column 'user_id' at row 1	

**Stored procedure- HEALTH\_STATUS**

This stored procedure will return the average steps counts, average sleep time, average calories, average distance and health status for a user, input parameter: UserID;

We have considered the below criteria to determine health Status.

Number of Steps - more than 1000 per day

Calories burnt - more than 2000

Sleep time: 7 to 9 hrs of sleep

Distance: 6 miles per day

Health Status – Healthy If User satisfies above conditions.

**With good Data:****SQL Code:**

```
USE fitbit_db;
```

```
DROP PROCEDURE IF EXISTS health_status;
```

```
DELIMITER //
```

```
CREATE PROCEDURE health_status
```

```
(
```

```

id_param BIGINT)
BEGIN

DECLARE avg_steps_var DOUBLE;
DECLARE no_of_records INT;
DECLARE      avg_distance_var DOUBLE;
DECLARE avg_calories_var DOUBLE;
DECLARE user_status TEXT;

DECLARE avg_sleeptimehrs DOUBLE;

SET avg_sleeptimehrs = get_avg_sleeptime(id_param);
SELECT
    ROUND(AVG(TotalSteps), 2),
    ROUND(AVG(TotalDistance), 2),
    ROUND(AVG(Calories), 2),
    COUNT(*)
INTO avg_steps_var , avg_distance_var , avg_calories_var , no_of_records FROM
    fitbit_db.dailyactivity
WHERE
    Id = id_param;

    IF(avg_steps_var > 1000) AND (avg_calories_var > 2000) AND ( avg_distance_var > 6 )
and ( avg_sleeptimehrs > 7 OR avg_sleeptimehrs < 9) THEN
        SET user_status = "Healthy";
    ELSE
        SET user_status = "Needs to work on Health";
    END IF;

SELECT
    avg_steps_var AS Average_StepsCount,
    avg_distance_var AS Avgerage_Distance,
    avg_calories_var AS Average_ColoriesBurn,
    avg_sleeptimehrs as Average_sleeptime,
    no_of_records,
        user_status AS Health_Status;

END//

DELIMITER ;

CALL health_status(1503960366);

```

**Results:**

48 • `CALL health_status(1503960366);`

Result Grid   Filter Rows:   Export:   Wrap Cell Content:						
	Average_StepsCount	Avgerage_Distance	Average_ColoriesBurn	Average_sleeptime	no_of_records	Health_Status
▶	12116.74	7.81	1816.42	6.21	31	Needs to work on Health

Result 1 x

Output

Action Output

#	Time	Action	Message
✓ 1	01:09:52	USE fitbit_db	0 row(s) affected
✓ 2	01:09:52	DROP PROCEDURE IF EXISTS health_status	0 row(s) affected
✓ 3	01:09:52	CREATE PROCEDURE health_status (id_param BIGINT...	0 row(s) affected
✓ 4	01:09:52	CALL health_status(1503960366)	1 row(s) returned

## With Bad Data: Passing incorrect number of arguments

### SQL Code:

`CALL health_status();`

### Results:

46 `DELIMITER ;`  
 47  
 48 • `CALL health_status();`

Output

Action Output

#	Time	Action	Message
✓ 1	01:13:11	USE fitbit_db	0 row(s) affected
✓ 2	01:13:11	DROP PROCEDURE IF EXISTS health_status	0 row(s) affected
✓ 3	01:13:11	CREATE PROCEDURE health_status (id_param BIGINT...	0 row(s) affected
✗ 4	01:13:11	CALL health_status()	Error Code: 1318. Incorrect number of arguments for PROCEDURE fitbit_db.health_status; expected