Insert Statement

The INSERT INTO statement

```
INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2,
value3, ...);
```

- If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. The INSERT INTO syntax would be as follows:

```
INSERT INTO table_name VALUES (value1, value2, value3, ...);
```

The INSERT INTO statement (examples)

```
* predjite na bazu vivify_blog

INSERT INTO posts (id, title, content, created_at, rating, category, teaser) VALUES
(101, 'Some Title', 'Beautiful content right here', '2017-12-22', 0, 'politics',
'Teaser');

INSERT INTO posts (id, title, content, created_at) VALUES (102, 'Some Other Title',
'Hell of a content', '2017-10-12');



INSERT INTO posts (id, title, content, created_at) VALUES (103, 'Different Title',
'Just a simple content', '2016-01-14');
```

The NULL value

- A field with a NULL value is a field with no value.

- If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value.

How to test for NULL values?

```
SELECT column_names FROM table_name WHERE column_name IS NULL;
```

```
SELECT column_names FROM table_name WHERE column_name IS NOT NULL;
```

- It is not possible to test for NULL values with comparison operators, such as =, <, or <>

The NULL value (example)

```sql
SELECT * FROM posts WHERE updated_at IS NULL;
```

```sql
SELECT * FROM posts WHERE updated_at IS NOT NULL;
```

Update Statement

The UPDATE Statement

```
UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;
```

- **Be careful when updating records in a table!** Notice the WHERE clause in the UPDATE statement. The WHERE clause specifies which record(s) should be updated. If you omit the WHERE clause, all records in the table will be updated!

The UPDATE Statement (examples)

```
UPDATE posts SET title = 'Novi naslov' WHERE id = 1;
```

```
UPDATE posts SET title = 'Jos noviji naslov', content = 'blah blah' WHERE id = 2;
```

```
UPDATE posts SET title = 'Originalni title', content = 'truc truc' WHERE id = 2 AND
title = 'Jos noviji naslov';
```

Delete Statement

The DELETE Statement

```
DELETE FROM table_name WHERE condition;
```

- **Be careful when deleting records in a table!** Notice the WHERE clause in the DELETE statement. The WHERE clause specifies which record(s) should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

The DELETE Statement (examples)

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

```
Test with: SELECT * FROM posts;
```

```
DELETE FROM posts WHERE updated_at IS NULL;
```