SQL INSERT (INSERT INTO / INSERT ALL)

SQL INSERT statement is used to inserting new records into database table. You can insert new record following different way,

- INSERT INTO Statement
- INSERT ALL Statement
- Multiple Row INSERT into TABLE Statement
- INSERT Data only in specified COLUMNS
- INSERT INTO SELECT Statement

INSERT INTO statement

Using INSERT INTO statement to insert record into database.

When inserting data into table no need to specify the column names if values is table structure wise (column wise).

Syntax

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

Example

```
SQL> INSERT INTO users_info VALUES (1, 'Opal Kole', '63 street Ct.', '000-444-7847');

1 row created.
```

When you inserting data into table and you haven't know table structure you must specify the column name.

Syntax

```
    INSERT INTO table_name [ (column_name1, column_name2, ...) ]
      VALUES (value1, value2, ...);
```

Example

```
SQL> INSERT INTO users_info (name, address, no, contact_no)
  VALUES ('Beccaa Moss', '2500 green city.', 3, '000-444-7142');

1 row created.
```

INSERT ALL statement

Using INSERT ALL statement to insert more then one records into table.

We can insert more then one record in single SQL INSERT statement.

Syntax

```
    INSERT ALL
        INTO table_name [ (column_name1, column_name2, ...) ] VALUES (record1_value1, record1_value2, ...)
        INTO table_name [ (column_name1, column_name2, ...) ] VALUES (record2_value1, record2_value2, ...)
        INTO table_name [ (column_name1, column_name2, ...) ] VALUES (record3_value1, record3_value2, ...)
        ....
        SELECT * FROM dual;
```

Example

```
SQL> INSERT ALL
  INTO users_info (no, name, address, contact_no) VALUES (4, 'Paul Singh', '1343 Prospect St', '000-444-7141')
  INTO users_info (no, name, address, contact_no) VALUES (5, 'Ken Myer', '137 Clay Road', '000-444-7084')
  INTO users_info (no, name, address, contact_no) VALUES (6, 'Jack Evans', '1365 Grove Way', '000-444-7957')
  INTO users_info (no, name, address, contact_no) VALUES (7, 'Reed Koch', '1274 West Street', '000-444-4784')
SELECT * FROM dual;

4 rows created.
```

SQL Multiple Row Insert into Table Statements

You can insert multiple record by this way first you execute INSERT INTO statement with & sign with column name. If you want to add another record you just execute forward slash (/) to again execute last statement automatically and you can insert new data again.

What is Forward Slash (/)?

Forward Slash (/) - Forward slash tell to a Oracle engine to execute last statement again.

```
SQL> INSERT INTO users_info VALUES (&no, &name, &address, &contact_no);
Enter value for no: 8
Enter value for name: 'Gabe Hee'
Enter value for address: '1220 Dallas Drive'
Enter value for contact_no: '000-444-4584'
old   1: INSERT INTO users_info VALUES (&no, &name, &address, &contact_no)
new   1: INSERT INTO users_info VALUES (8, 'Gabe Hee', '1220 Dallas Drive', '000-444-4584')

1 row created.

SQL> /
Enter value for no: 9
Enter value for name: 'Ben Mares'
Enter value for address: '101 Candy Road'
Enter value for contact_no: '000-444-5484'
old   1: INSERT INTO users_info VALUES (&no, &name, &address, &contact_no)
new   1: INSERT INTO users_info VALUES (9, 'Ben Mares', '101 Candy Road', '000-444-5484')

1 row created.

SQL>
```

SQL Insert Data only in specified COLUMNS

You can insert data only specific column. When you write INSERT statement you have to specify column name to inserting only that column data into table.

Syntax

```
INSERT INTO Table_Name (specific_column_name1, ...)
    VALUES (value1,...);
```

Example

```
SQL> INSERT INTO users_info(no, name) VALUES (10, 'Sariya Vargas');

1 row created.
```

INSERT INTO SELECT Statement

INSERT INTO SELECT Statement to insert data that data you are getting from another table.

Syntax

```
 INSERT INTO new_table_name [(column_name1,column_name2,...)]
     SELECT column_name1, column_name1 ... FROM
     another_table_name
     [WHERE condition];
```

Example

```
SQL> CREATE TABLE demo_tbl(
     no NUMBER(3),
     name VARCHAR2(50)
     );

Table created.

SQL> INSERT INTO demo_tbl (no, name)
     SELECT no, name FROM
     users_info;

10 rows created.
```