# COALESCE() - Preamble

# COALESCE() - Preamble

Here we will study something a bit more sophisticated.

IF NULL() and COALESCE() are among the advanced SQL functions in the toolkit of SQL professionals. They are used when null values are dispersed in your data table and you would like to substitute the null values with another value.

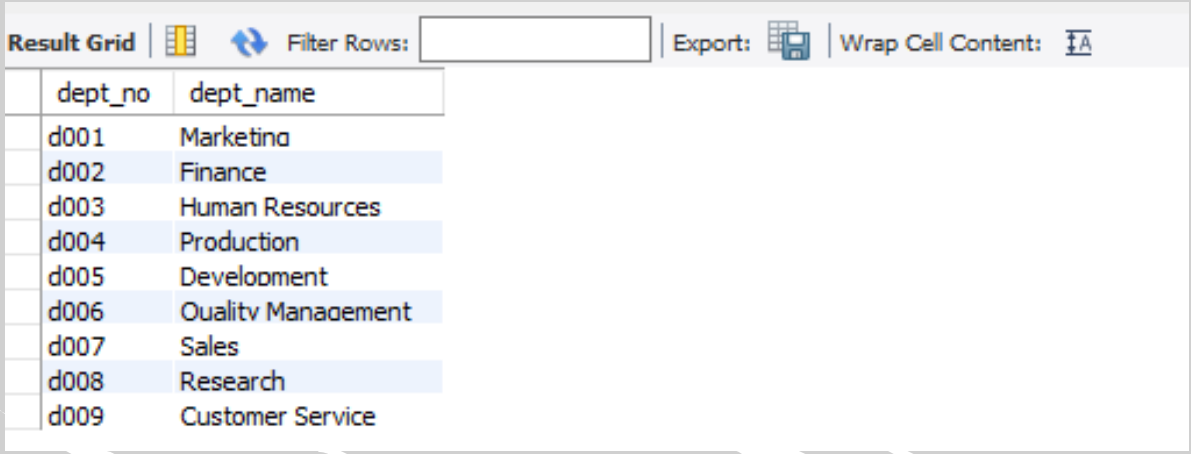
So, let's adjust the "Departments" duplicate in a way that suits the purposes of the next video, in which we will work with IF NULL() and COALESCE().

First, let's look at our table and see what we have there.

# COALESCE() - Preamble

SQL

```
SELECT * FROM departments_dup;
```



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

| dept_no | dept_name |
|---|---|
| d001 | Marketing |
| d002 | Finance |
| d003 | Human Resources |
| d004 | Production |
| d005 | Development |
| d006 | Quality Management |
| d007 | Sales |
| d008 | Research |
| d009 | Customer Service |

Nine departments, with their department numbers and names provided. Ok!

# COALESCE() - Preamble

Currently, as shown in the DDL statement of this table, the "Department name" field is with a NOT NULL constraint, which naturally means we must insert a value in each of its rows.

DDL for employees.departments_dup

```
1  CREATE TABLE `departments_dup` (
2      `dept_no` char(4) NOT NULL,
3      `dept_name` varchar(40) NOT NULL
4  ) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

# COALESCE() - Preamble

Now, with the ALTER TABLE statement and the CHANGE COLUMN command, we will modify this constraint and allow null values to be registered in the “department name” column.


```sql
ALTER TABLE departments_dup

CHANGE COLUMN dept_name dept_name VARCHAR(40) NULL;
```

# COALESCE() - Preamble

Right after that, we will insert into the department number column of this table a couple of data values – D-10 and D-11, the numbers of the next two potential departments in the “Departments Duplicate” table.


```sql
INSERT INTO departments_dup(dept_no) VALUES ('d010'), ('d011');
```

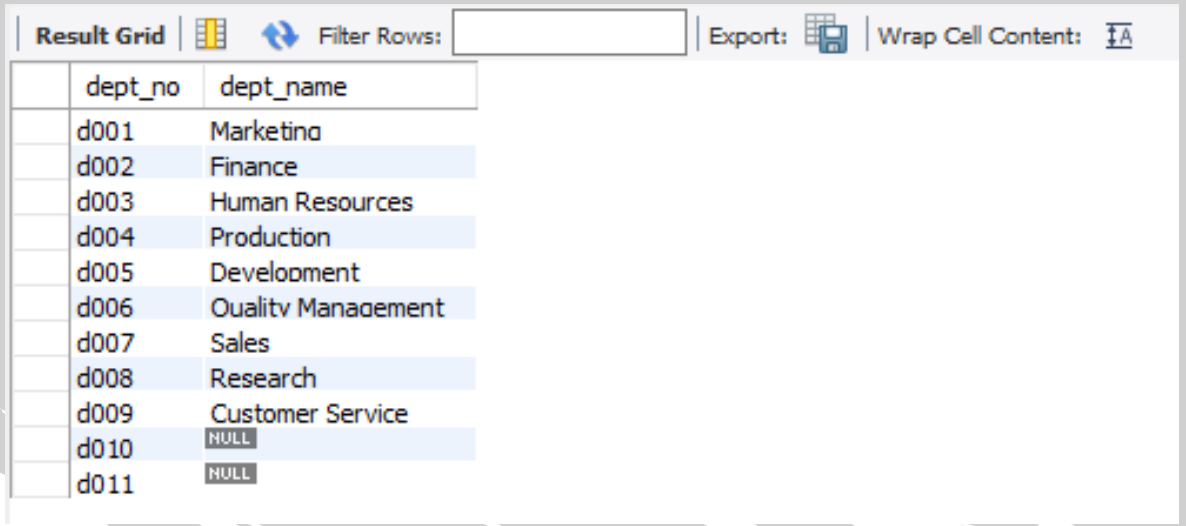
# COALESCE() - Preamble

By running this SELECT query over here, you can see whether this operation was carried out successfully.


```sql
SELECT
    *
FROM
    departments_dup
ORDER BY dept_no ASC;
```

# COALESCE() - Preamble

We have the two new department numbers listed below, and in the "Department name" column we can see two null values. The latter happened because we allowed for null values to exist in this field, "Department name". Thus, Workbench will indicate that a value in a cell is missing by attaching a "null" label to it. Great!




| dept_no | dept_name |
|---|---|
| d001 | Marketing |
| d002 | Finance |
| d003 | Human Resources |
| d004 | Production |
| d005 | Development |
| d006 | Quality Management |
| d007 | Sales |
| d008 | Research |
| d009 | Customer Service |
| d010 | NULL |
| d011 | NULL |

# COALESCE() - Preamble

The next adjustment we'll have to make is adding a third column called "Department manager". It will indicate the manager of the respective department. For now, we will leave it empty, and will add the NULL constraint. Finally, we will place it next to the "Department name" column by typing "AFTER "Department name".


```sql
ALTER TABLE employees.departments_dup

ADD COLUMN dept_manager VARCHAR(255) NULL AFTER dept_name;
```

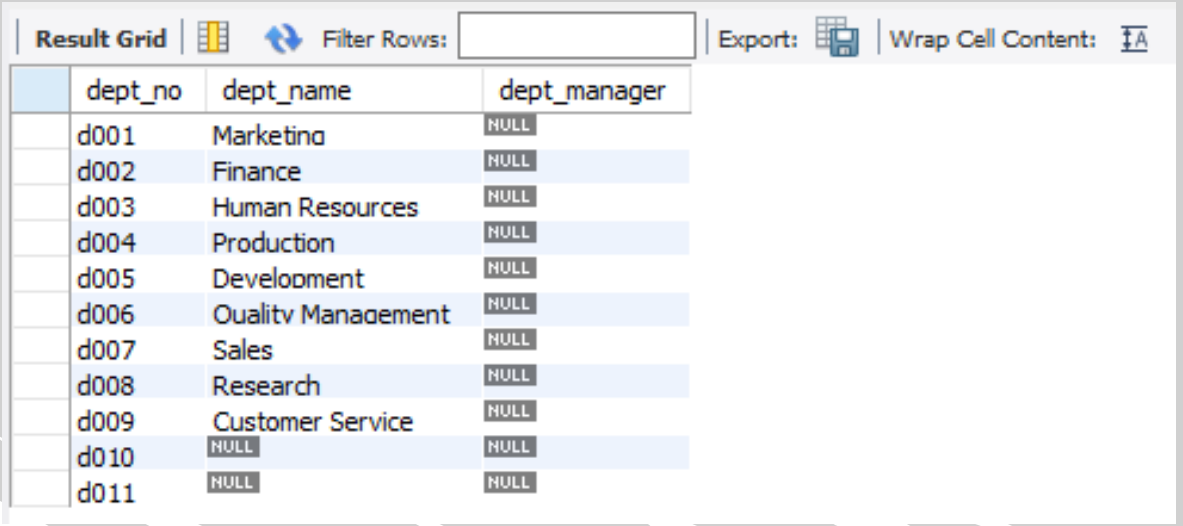
# COALESCE() - Preamble

Let’s check the state of the “Departments duplicate” table now.


```sql
SELECT
    *
FROM
    departments_dup
ORDER BY dept_no ASC;
```

# COALESCE() - Preamble

Exactly as we wanted, right? The third column is completely empty and we have null values in the last two records. These are the “department name” and “manager” fields.




| dept_no | dept_name | dept_manager |
|---|---|---|
| d001 | Marketing | NULL |
| d002 | Finance | NULL |
| d003 | Human Resources | NULL |
| d004 | Production | NULL |
| d005 | Development | NULL |
| d006 | Quality Management | NULL |
| d007 | Sales | NULL |
| d008 | Research | NULL |
| d009 | Customer Service | NULL |
| d010 | NULL | NULL |
| d011 | NULL | NULL |

# COALESCE() - Preamble

To save the “Departments duplicate” table in its current state, execute a COMMIT statement.


```
COMMIT;
```

Here we’ll end the setup for the video about IF NULL() and COALESCE().

# Good luck!