

## BrightLearn Tutorials

### SQL NULL Functions Exercise: SOLUTIONS

Q1. Show all employees with their salary. If salary is NULL, display 0.

#### SQL QUERY:

```
SELECT employee_id, name,  
       IFNULL(salary, 0) AS salary_with_default  
FROM Employees;
```

#### Expected output:

| employee_id | name    | salary_with_default |
|-------------|---------|---------------------|
| 1           | Alice   | 5000                |
| 2           | Bob     | 0                   |
| 3           | Charlie | 7000                |
| 4           | Dana    | 0                   |

Q2. Show employee names with their department. If department is NULL, show "Not Assigned".

#### SQL QUERY:

```
SELECT employee_id, name,  
       COALESCE(department, 'Not Assigned') AS department_name  
FROM Employees;
```

#### Expected output:

| employee_id | name    | department_name |
|-------------|---------|-----------------|
| 1           | Alice   | HR              |
| 2           | Bob     | IT              |
| 3           | Charlie | Not Assigned    |
| 4           | Dana    | Finance         |

Q3. Find orders with NULL customer\_id using ISNULL().

#### SQL QUERY:

```
SELECT order_id, customer_id  
FROM Orders  
WHERE ISNULL(customer_id);
```

#### Expected output:

| order_id | customer_id |
|----------|-------------|
| 103      | NULL        |

Q4. Show all orders. If delivery\_date is NULL, show 'Pending'.

**SQL QUERY:**

SELECT order\_id, customer\_id,

COALESCE(delivery\_date, 'Pending') AS delivery\_status

FROM Orders;

**Expected output:**

| order_id | customer_id | delivery_status |
|----------|-------------|-----------------|
| 101      | 201         | 2024-12-01      |
| 102      | 202         | Pending         |
| 103      | NULL        | 2024-12-03      |

Q5. Show all students and their grades. Replace NULL with 0.

**SQL QUERY:**

SELECT student\_id, name,

IFNULL(grade, 0) AS final\_grade

FROM Students;

**Expected output:**

| student_id | name   | final_grade |
|------------|--------|-------------|
| 1          | Ethan  | 85          |
| 2          | Maya   | 0           |
| 3          | Olivia | 90          |

Q6. Count students who haven't been graded.

**SQL QUERY:**

SELECT COUNT(\*) AS not\_graded\_count

FROM Students

WHERE grade IS NULL;

**Expected output:**

| not_graded_count |
|------------------|
| 1                |

Q7. Show product name, price, and final price after discount (assume 0 if discount is NULL).

**SQL QUERY:**

SELECT product\_id, name,

price - IFNULL(discount, 0) AS final\_price

FROM Products;

**Expected output:**

| product_id | name     | final_price |
|------------|----------|-------------|
| 501        | Keyboard | 25          |
| 502        | Mouse    | 10          |
| 503        | Monitor  | 100         |

Q8. Count how many customers have no email.

**SQL QUERY:**

SELECT COUNT(\*) AS missing\_email\_count

FROM Customers

WHERE email IS NULL;

**Expected output:**

| missing_email_count |
|---------------------|
| 2                   |

Q9. Show all customers with email. If NULL, display "No Email".

**SQL QUERY:**

SELECT customer\_id, name,

COALESCE(email, 'No Email') AS email\_display

FROM Customers;

**Expected output:**

| customer_id | name   | email_display  |
|-------------|--------|--|
| 1           | Linda  | No Email   |
| 2           | Joseph | <a href="mailto:joseph@mail.com">joseph@mail.com</a> |
| 3           | Nia    | No Email   |

Q10. Show payment details with method replaced by "Unknown" if NULL.

**SQL QUERY:**

SELECT payment\_id,

COALESCE(method, 'Unknown') AS method\_display,

status

FROM Payments;

**Expected output:**

| payment_id | method_display | status  |
|------------|----------------|---------|
| 301        | Credit         | NULL    |
| 302        | PayPal         | Success |
| 303        | Unknown        | Failed  |

Q11. Show items and their quantity (0 if NULL).

**SQL QUERY:**

SELECT item\_id, item\_name,

IFNULL(quantity, 0) AS quantity\_checked

FROM Inventory;

**Expected output:**

| item_id | item_name | quantity_checked |
|---------|-----------|------------------|
| 1       | Pen       | 0                |
| 2       | Notebook  | 150              |
| 3       | Eraser    | 0                |

Q12. Show employee ID and the first available value among bonus or commission.

**SQL QUERY:**

SELECT emp\_id,

COALESCE(bonus, commission, 0) AS first\_available\_reward

FROM Employees\_Extra;

**Expected output:**

| emp_id | first_available_reward |
|--------|------------------------|
| 1      | 300                    |
| 2      | 100                    |
| 3      | 0                      |

Q13. Count classes that don't have a room assigned.

**SQL QUERY:**

SELECT COUNT(\*) AS no\_room\_count

FROM Classes

WHERE room IS NULL;

**Expected output:**

| no_room_count |
|---------------|
| 2             |

Q14. Show attendance records with status. Replace NULL with "Not Marked".

**SQL QUERY:**

```
SELECT student_id, date,
```

```
COALESCE(status, 'Not Marked') AS attendance_status
```

```
FROM Attendance;
```

**Expected output:**

| student_id | date       | attendance_status |
|------------|------------|-------------------|
| 1          | 2025-04-01 | Not Marked        |
| 2          | 2025-04-01 | Present           |
| 3          | 2025-04-01 | Absent            |

Q15. Show account ID, account\_type (or 'Unknown'), and balance (or 0).

**SQL QUERY:**

```
SELECT account_id,
```

```
COALESCE(account_type, 'Unknown') AS type_display,
```

```
IFNULL(balance, 0) AS balance_checked
```

```
FROM Bank_Accounts;
```

**Expected output:**

| account_id | type_display | balance_checked |
|------------|--------------|-----------------|
| A1         | Savings      | 0               |
| A2         | Current      | 5000            |
| A3         | Unknown      | 2000            |

Q16. Show all projects with a start date. If start\_date is NULL, display 'TBD'.

**SQL QUERY:**

```
SELECT project_id, title,
```

```
COALESCE(start_date, 'TBD') AS start_display
```

```
FROM Projects;
```

**Expected output:**

| project_id | title          | start_display |
|------------|----------------|---------------|
| 1          | Website Revamp | 2025-01-10    |
| 2          | Mobile App     | TBD           |
| 3          | Data Migration | TBD           |

Q17. Display reviews showing comment (or 'No Comment') and rating (or 0).

**SQL QUERY:**

```
SELECT review_id, product_id,
```

```
    COALESCE(comment, 'No Comment') AS comment_display,
```

```
    IFNULL(rating, 0) AS rating_display
```

```
FROM Reviews;
```

**Expected output:**

| review_id | product_id | comment_display | rating_display |
|-----------|------------|-----------------|----------------|
| 1         | 501        | Great product   | 4              |
| 2         | 502        | No Comment      | 0              |
| 3         | 503        | Works fine      | 3              |

Q18. Show the supplier contact number. Use COALESCE(phone, alt\_phone, 'No Contact').

**SQL QUERY:**

```
SELECT supplier_id, name,
```

```
    COALESCE(phone, alt_phone, 'No Contact') AS contact_number
```

```
FROM Suppliers;
```

**Expected output:**

| supplier_id | name          | contact_number |
|-------------|---------------|----------------|
| 1           | Global Goods  | 123456789      |
| 2           | Best Supplies | 987654321      |
| 3           | ValueSource   | No Contact     |

Q19. Show all users and their preferences. Replace all NULLs with defaults:

**SQL QUERY:**

```
SELECT user_id,
```

```
    COALESCE(theme, 'Light') AS theme_set,
```

```
    COALESCE(language, 'English') AS language_set,
```

```
    COALESCE(timezone, 'UTC') AS timezone_set
```

```
FROM User_Settings;
```

**Expected output:**

| user_id | theme_set | language_set | timezone_set |
|---------|-----------|--------------|--------------|
| 1       | Light     | English      | UTC          |
| 2       | Dark      | English      | UTC+1        |
| 3       | Light     | English      | UTC          |

Q20. Show maintenance log with:

**SQL QUERY:**

```
SELECT record_id, machine_id,
```

```
COALESCE(issue, 'Unknown Issue') AS issue_log,
```

```
COALESCE(technician, 'Not Assigned') AS technician_name
```

```
FROM Maintenance;
```

**Expected output:**

| record_id | machine_id | issue_log     | technician_name |
|-----------|------------|---------------|-----------------|
| 1         | M101       | Overheating   | Not Assigned    |
| 2         | M102       | Unknown Issue | Not Assigned    |
| 3         | M103       | Jammed        | Alex            |