1. **Trigger to Automatically Update Total Price Upon Insertion of Items:**
   Table: **Order_Items**

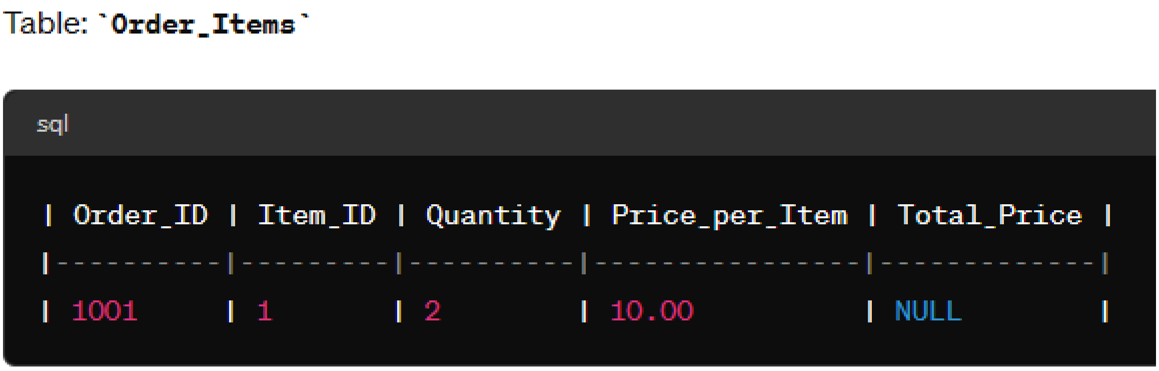


Table: `Order_Items`

sql

```
| Order_ID | Item_ID | Quantity | Price_per_Item | Total_Price |
|----------|---------|----------|----------------|-------------|
| 1001     | 1       | 2        | 10.00          | NULL        |
```

Trigger:

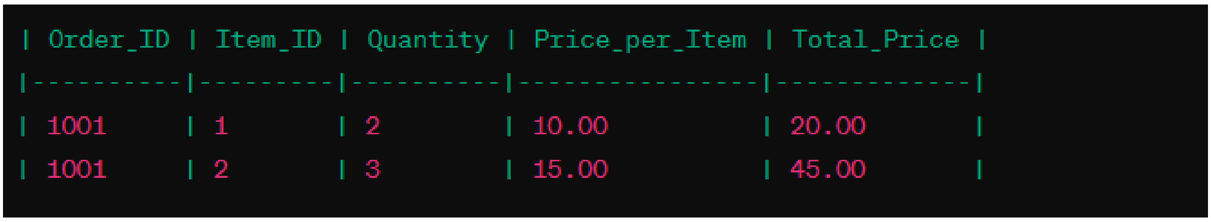
```
CREATE TRIGGER update_total_price

AFTER INSERT ON Order_Items

FOR EACH ROW

BEGIN

    UPDATE Order_Items

    SET Total_Price = NEW.Quantity * NEW.Price_per_Item

    WHERE Order_ID = NEW.Order_ID;

END;
```

Output:

After inserting a new item:

INSERT INTO Order_Items (Order_ID, Item_ID, Quantity, Price_per_Item) VALUES (1001, 2, 3, 15.00);

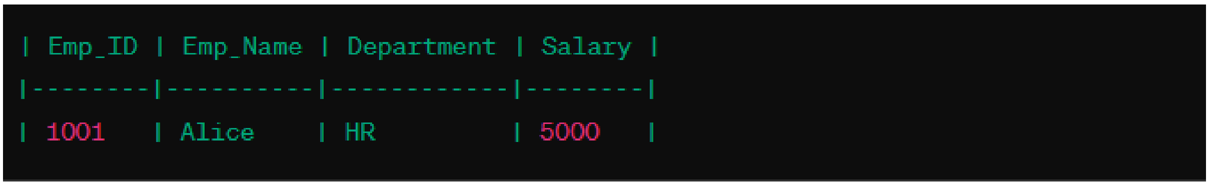
Table after insertion:



```
| Order_ID | Item_ID | Quantity | Price_per_Item | Total_Price |
|----------|---------|----------|----------------|-------------|
| 1001     | 1       | 2        | 10.00          | 20.00       |
| 1001     | 2       | 3        | 15.00          | 45.00       |
```

2. **Trigger to Prevent Deletion of Records with Specific Conditions:**

   Table: **Employees**



```
| Emp_ID | Emp_Name | Department | Salary |
|--------|----------|------------|--------|
| 1001   | Alice    | HR         | 5000   |
```

Trigger:

CREATE TRIGGER prevent_hr_deletion

BEFORE DELETE ON Employees

FOR EACH ROW

BEGIN

  IF OLD.Department = 'HR' THEN

    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot delete HR employees.';

  END IF;

END;

Output:

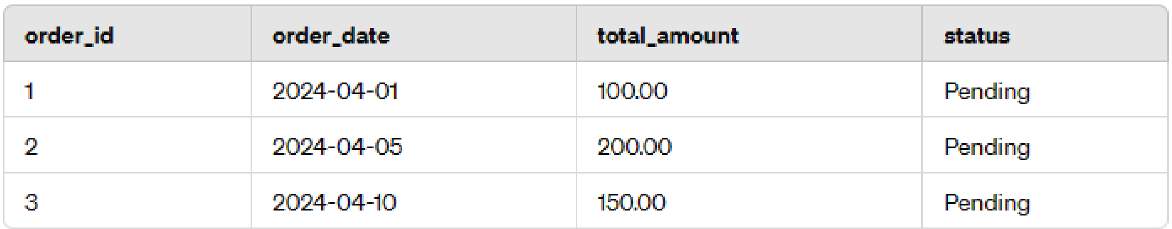
Attempting to delete an HR employee:

DELETE FROM Employees WHERE Emp_ID = 1001;

Output:

Error Code: 1644. Cannot delete HR employees.

3. Consider a table named **orders** with the following structure:



order_id	order_date	total_amount	status
1	2024-04-01	100.00	Pending
2	2024-04-05	200.00	Pending
3	2024-04-10	150.00	Pending

After Update Trigger to Change Status:
Assuming we have a trigger that changes the status of an order to "Completed" when the total amount exceeds 200.

CREATE TRIGGER update_order_status_after_update

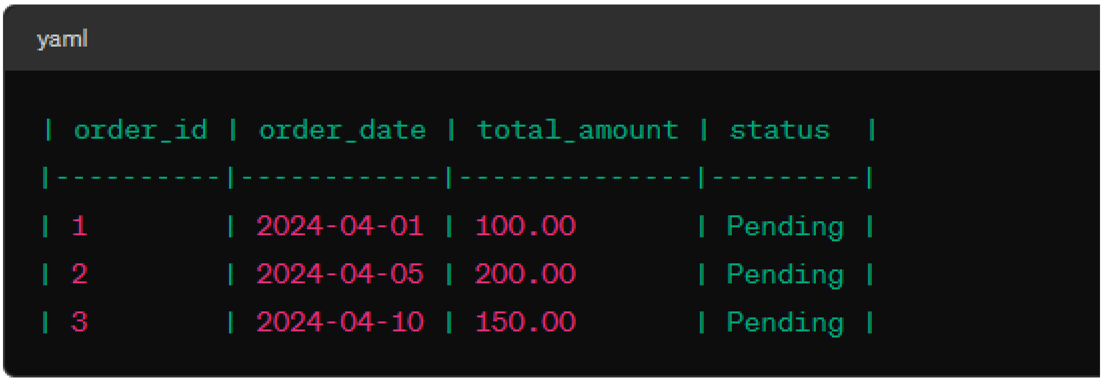
AFTER UPDATE ON orders

```
FOR EACH ROW
BEGIN
    IF NEW.total_amount > 200 THEN
        SET NEW.status = 'Completed';
    END IF;
END;
```

Before Update:

```
SELECT * FROM orders;
```

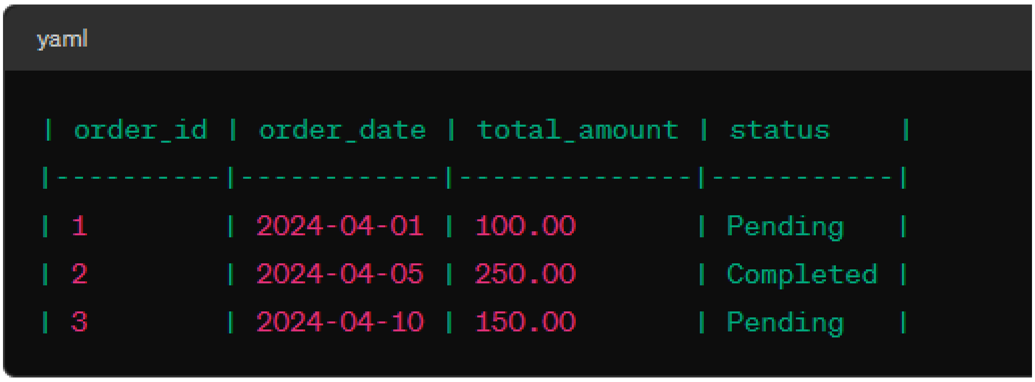
Output:



```
| order_id | order_date | total_amount | status  |
|----------|------------|--------------|---------|
| 1        | 2024-04-01 | 100.00       | Pending |
| 2        | 2024-04-05 | 200.00       | Pending |
| 3        | 2024-04-10 | 150.00       | Pending |
```

After Update (Changing total_amount of order_id 2 to 250):

```
UPDATE orders SET total_amount = 250 WHERE order_id = 2;
SELECT * FROM orders;
```



```
| order_id | order_date | total_amount | status    |
|----------|------------|--------------|-----------|
| 1        | 2024-04-01 | 100.00       | Pending   |
| 2        | 2024-04-05 | 250.00       | Completed |
| 3        | 2024-04-10 | 150.00       | Pending   |
```

4. **Before Insert Trigger to Calculate Total Amount:**

Assuming we have a trigger that calculates the total amount of an order before insertion.

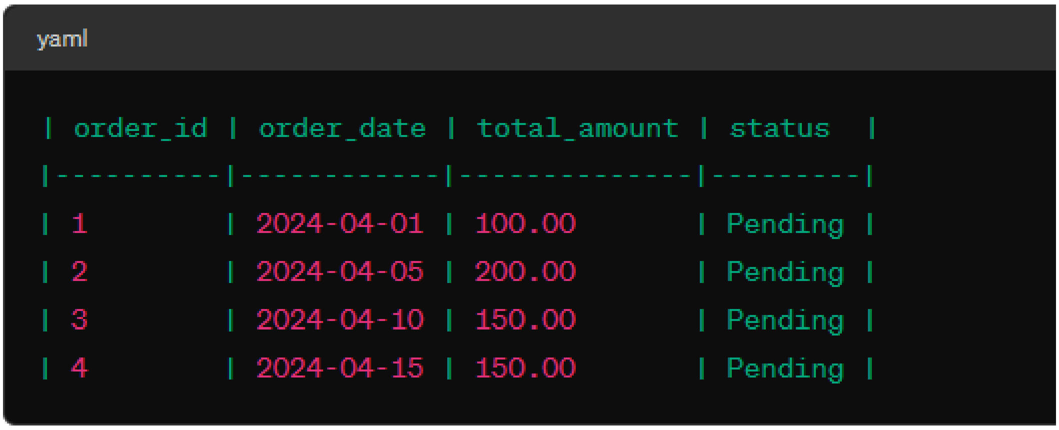
```
CREATE TRIGGER calculate_total_amount_before_insert
BEFORE INSERT ON orders
```

```
FOR EACH ROW
BEGIN
    SET NEW.total_amount = NEW.product_price * NEW.quantity;
END;
```

Before Insert:

```
INSERT INTO orders (order_date, product_price, quantity, status) VALUES ('2024-04-15', 50.00, 3, 'Pending');
SELECT * FROM orders;
```

Output:



```
| order_id | order_date | total_amount | status   |
|----------|------------|--------------|---------|
| 1        | 2024-04-01 | 100.00       | Pending |
| 2        | 2024-04-05 | 200.00       | Pending |
| 3        | 2024-04-10 | 150.00       | Pending |
| 4        | 2024-04-15 | 150.00       | Pending |
```

5. **Before Delete Trigger to Prevent Deletion:**

Assuming we have a trigger that prevents the deletion of orders with a status of "Completed".

```
CREATE TRIGGER prevent_delete_completed_order
BEFORE DELETE ON orders
FOR EACH ROW
BEGIN
    IF OLD.status = 'Completed' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot delete completed orders.';
    END IF;
END;
```

Before Delete:

```
DELETE FROM orders WHERE order_id = 2;
SELECT * FROM orders;
```

Output:

```
Error: Cannot delete completed orders.
```