Exercise 3: SQL CASE Statements

1. Products

Product_id	product_name	price	Price_category
1	Laptop	1200.00	
2	Phone	800.00	
3	Keyboard	45.00	
4	Monitor	300.00	
5	Mouse	25.00	

Qn

- Classify each product by price:
- 'Expensive' if price between 100 and 1000
- 'Budget' if price < 100

```
SELECT product_id,
       Product_name,
       Price,
CASE
    WHEN price < 100 THEN 'Budget'
    WHEN price BETWEEN 100 AND 1000 THEN 'mid_range'
    WHEN price > 1000 THEN 'Expensive'
    END AS price_category
FROM products;
```

2. Label each order:

- 'High Value' for orders >= 1000
- 'Medium Value' for 500 - 999.99
- 'Low Value' for orders < 500

customer name	amount	order_value_category
Alice	150.00	low value
Bob	560.00	medium value
charlie	999.99	medium value
Diana	45.50	low value
Ethan	1200.00	High value

```
SELECT customer_name, amount,
CASE
    WHEN amount < 500 THEN 'Low Value'
    WHEN amount BETWEEN 500 AND 999.99 THEN 'medium Value'
    WHEN amount >= 1000 THEN 'High value'
    END AS order_value_category
FROM orders
```

3. Categorize employee position:
- If in 'IT' and salary > 80000 → 'Senior IT'
- If in 'HR' and salary > 55000 → 'Experienced HR'
- Otherwise → 'Staff'

emp_name	department	salary	position_level
John	IT	85000	Senior IT
Sara	HR	60000	Experienced HR
Mark	IT	75000	Staff
Lucy	Finance	95000	Staff
Tom	HR	55000	Staff

```
SELECT emp_name, department,
       salary,
CASE
  WHEN department = 'IT' AND salary > 80000 THEN 'Senior IT'
  WHEN department = 'HR' AND salary > 55000 THEN 'experienced HR'
  ELSE 'staff'
END AS position_level
FROM employees;
```

4. Assign a letter grade:
- >= 90: 'A'
- 80-89: 'B'
- 70-79: 'C'
- 60-69: 'D'
- < 60: 'F'

student_name	score	grade
Anna	92	A
Ben	76	C
Cara	59	F
David	83	B
Ella	68	D

```
SELECT student_name,
       score,
CASE
  WHEN score >= 90 THEN 'A'
  WHEN score BETWEEN 80 AND 89 THEN 'B'
  WHEN score BETWEEN 70 AND 79 THEN 'C'
  WHEN score BETWEEN 60 AND 69 THEN 'D'
  ELSE 'F'
END AS grade
FROM students
```

5 Label delivery performance:

- ≤ 30 mins: 'Fast'
- 31 – 60 mins: 'On Time'
- 60 mins: 'Late'

```
SELECT delivery_id,
       delivery_time_minutes,
CASE
  WHEN delivery_time_minutes <= 30 THEN 'Fast'
  WHEN delivery_time_minutes BETWEEN 31 AND 60 THEN 'On Time'
ELSE 'Late'
END AS performance
FROM deliveries;
```

delivery_id	delivery_time_minutes	performance
1	45	On Time
2	80	Late
3	30	Fast
4	65	Late
5	100	Late

6 Convert priority to labels

- 3 → 'High'
- 2 → Medium
- 1 → 'Low'

```
SELECT
issue_type,
priority,
CASE
  WHEN priority = 3 THEN 'High'
  WHEN priority = 2 THEN 'Medium'
WHEN priority = 1 'THEN 'Low'
END AS priority_label
FROM tickets;
```

Issue_type	priority	priority_label
Login issue	1	Low
Server down	3	High
Slow system	2	Medium
Email error	2	Medium
Password reset	1	Low

7. Calculate attendance % and classify:
- >= 90% → 'Excellent'
- 75-89% → 'Good'
- < 75% → 'Needs Improvement'

```
SELECT student_id,
  (days_present * 100 / total_days) AS attendance_percentage,
CASE
  WHEN attendance_percentage >= 90 THEN 'Excellent'
  WHEN attendance_percentage BETWEEN 75 AND 89 THEN 'Good'
ELSE 'Needs Improvement'
END AS attendance_status
FROM attendance;
```

student_id	attendance_percentage	attendance_status
1	90.0	Excellent
2	60.0	Needs improvement
3	96.0	Excellent
4	50.0	Needs improvement
5	100.0	Excellent

8. Label stock status:
- 0 → 'Out of stock'
- 1-5 → 'Low Stock'
- 5 → 'In stock'

```
SELECT product_id,
       stock_qty,
CASE
  WHEN stock_qty = 0 THEN 'Out of stock'
  WHEN stock_qty BETWEEN 1 AND 4 THEN 'Low stock'
  ELSE 'In stock'
END AS stock_status
FROM products_inventory;
```

product_id	stock_qty	stock_status
1	5	In stock
2	0	out of stock
3	25	In stock
4	10	In stock
5	3	low stock

9. Classify by size
- ≥ 25 → 'Large'
- 10-24 → 'medium'
- < 10 → 'Small'

```
SELECT subject,
       enrolled students,
CASE
  WHEN enrolled_studet >= 25 THEN 'Large'
  WHEN enrolled_studnts BETWEEN 10 AND 24 THEN 'Medium'
  Else 'Small'
END AS class_size_category
FROM classes;
```

subject	enrolled_students	class_size_category
Math	30	Large
English	25	Large
Science	15	Medium
Art	5	small
History	20	Medium

10. Apply discount flag:
- If payment_method = 'cash' and amount >= 200 → eligible for discount
- Otherwise → 'not Eligble'

```
SELECT payment_id,
       payment_method
       amount,
CASE
  WHEN payment_method = 'Cash' AND amount >= 200 THEN 'eligible for discount'
  Else ~~EN~~ 'Not eligble'
END AS discount_eligibility
FROM payments;
```

payment_id	payment_method	amount	discount_eligibility
1	Card	50.00	Not eligible
2	Cash	200.00	eligible for discount
3	Card	150.00	Not eligible
4	Paypal	75.00	Not eligible
5	Cash	300.00	Eligible for discount