Exercise 3:

```
SELECT 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;
```

Product_name	Price	Price - category
Laptop	1200.00	Expensive
Phone	800.00	mid-range
keyboard	45.00	Budget
Monitor	300.00	mid-range
Mouse	25.00	Budget

2.
```
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,
```

Customer_name	amount	order value cat
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

3. SELECT emp_name,
department,
Salary
CASE
WHEN department = 'IT' AND Salary > 80000
WHEN department = 'HR' AND Salary > 55000
ELSE 'Staff'
END AS position_level
FROM employees;

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

4. SELECT Student-name,
score,
CASE WHEN score < 60 THEN 'F'
WHEN Score BETWEEN 60 AND 69 THEN 'D'
WHEN Score BETWEEN 70 AND 79 THEN 'C'
WHEN Score BETWEEN 80 AND 89 TEN 'B'
WHEN score ≥ 90 THEN 'A)
END AS grade
FROM Students;

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

5. SELECT delivery_id,
delivery_time_minutes
CASE WHEN dtm ≤ 30 mins THEN 'Fast'
WHEN dtm BETWEEN 31 AND 60 mins THEN 'on time'
WHEN dtm 60 mins THEN LATE
END AS performance
FROM deliveries

delivery_id	dtm	performance
1	45	On time
2	80	Late
3	30	fast
4	65	Late
5	100	Late

6. SELECT issue_type,
Priority
CASE
WHEN priority = 1 THEN 'Low'
WHEN priority = 2 THEN 'Medium'
WHEN priority = 3 THEN 'High'
END AS priority_level
FROM tickets;

issue_type	priority	Priority_level
login_issue	1	Low
Server down	3	High
Slow system	2	Medium
Email error	2	Medium
Password reset	1	Low

7. 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'
WHEN attendance_percentage < 75 THEN 'Needs improvement'
END AS attendance_status
FROM attendance;

Student_id	attendance_percentage	attendance_status
1	90	Excellent
2	60	Needs Improvement
3	96	Excellent
4	50	Need Improvement
5	100	Excellent

8. SELECT product_id,
stock_qty

CASE
WHEN stock_qty = 0 THEN 'Out of stock'
WHEN stock_qty BETWEEN 1 AND 5 THEN 'Low stock'
ELSE 'In stock'
END AS Stock_Status
FROM product_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. SELECT subject,
enrolled_students

CASE
WHEN enrolled_students < 10 THEN 'Small'
WHEN enrolled_students BETWEEN 10 AND 24 THEN 'Medium'
WHEN enrolled_students >= 25 THEN 'Large'
END AS class_size_category
FROM classes;

Subject	enrolled_students	class_size_category
Maths	30	Large
English	25	Large
Science	15	Medium
Art	5	Small
History	20	Medium

10. SELECT payment_id,
payment_method,
amount
CASE
WHEN payment_method = 'cash' AND amount >= 200 THEN 'Eligible for discount'
ELSE 'Not Eligible'
END AS discount_eligibility
FROM payments;

payment_id	payment_method	amount	discount_eligibility
1	Card	50	Not Eligible
2	cash	200	Eligible for discount
3	Card	150	Not eligible
4	Paypal	75	Not Eligible
5	Cash	300	Eligible for discount

JOIN Exercise 4.

①
```
SELECT Student_id,
       Student_name,
       grade
FROM Students AS A
INNER JOIN grades AS B
ON A.student_id = B.Student_id
```

Student_id	Student_name	grade
2	Bob	B
3	Charlie	A

②
```
SELECT emp_id,
       emp_name,
       dept_name
FROM employees AS A
LEFT JOIN departments AS B
ON A.emp_id = B.emp_id;
```

emp_id	emp_name	dept_name
1	John	NULL
2	LISa	HR
3	Mike	NULL

③
```
SELECT product_id,
       product_name,
       quantity
FROM products AS A
FULL OUTER JOIN Sales AS B
ON A.product_id = B.product_id
```

Product_id	product_name	quantity
1	Laptop	NULL
2	Mouse	50
3	Keyboard	NULL
4	NULL	30

④
```
SELECT order_id,
       customer_id,
       amount,
       customer_name
CASE
    WHEN customer_name IS NULL THEN 'NEW customer'
    ELSE 'Returning Customer'
END AS customer_type
FROM orders AS A
LEFT JOIN customers AS B
ON A.customer_id = B.customer_id;
```

order_id	customer_id	amount	customer_name
1			
2			
3			

order_id	customer_id	amount	customer_name	customer_type
1	101	500	Paul	Returning
2	102	300	Sarah	Returning
3	105	0	NULL	New customer

⑤
```
SELECT region_id,
       region_name,
       SUM(amount) AS total_sales
FROM Sales AS A
LEFT JOIN regions AS B
ON A.region_id = B.region_id
GROUP BY region_id;
```

region_id	region_name	total_sales
1	North	2000
2	South	3500
4	NULL	1000

(consult)

⑥
```
SELECT student_id,
       name,
       days_present
CASE
    WHEN days_present IS NULL THEN 'Poor att'
    WHEN days_present < 10 THEN 'Needs impro
    WHEN days_present > 10 THEN 'Excellent'
END AS attendance_status
FROM Students AS A
LEFT JOIN attendance AS B
ON A.student_id = B.student_id;
```

Student_id	name	days_p	attendance_status
1	Alice	18	Excellent
2	Bob	5	Needs improvement
3	Charlie	NULL	Poor attendance

(No record)

⑦
```
SELECT project_id,
       name,
       COUNT(task_id) AS task_count
GROUP BY project_id
FROM projects AS A
INNER JOIN tasks AS B
ON A.project_id = B.project_id;
```

Project_id	name	task_count
1	AI Chatbot	2
2	Website	1
4		

[consult]

⑧
```
SELECT cust_id,
       order_total,
       return total
CASE WHEN return total = NULL THEN 'No Return'
     WHEN return total > 0 THEN 'Returned'
END AS return_status
FROM orders AS A
FULL OUTER JOIN returns AS B
ON A.cust_id = B.cust_id;
```

cust_id	order_total	return total	return_status
11	120	20	Returned
12	250	NULL	No return
13	180	NULL	No return
14	NULL	100	Returned (No record)

(#) (9)

```
SELECT user_id,
       name,
    COUNT(user_id) AS login_count
FROM users AS A
LEFT JOIN logins AS B
ON A.user_id = B.user_id
ORDER BY login_count DESC
```

user_id	name	login_count
3	Steve	1
2	Gloria	2
1	Nelson	NULL

(10)

(a) · SELECT teacher_id
teacher_name
CASE S'
WHEN ~~Sub~~ Subject_name IS NULL THE NOSub
END AS subject_name
~~FROM A.teacher_id = B teacher_id~~
FROM teachers AS A
LEFT JOIN subjects AS B
ON A.teacher_id = B.teacher_id
Order By teacher_name ASC

teacher_id	teacher_name	subject name
1	Mr Hlongwane	Maths
2	Ms. Ndabal	NOSubject ASS
3	Mr. Dlamini	NO Subject ASS
1	Mr Hlongwane	Science