

## TASK-4 Use SQL queries to extract and analyse data from a database.

```
-- 1. Simple SELECT & WHERE
• SELECT
    customer_id,
    CONCAT(first_name, ' ', last_name) AS name,
    country
FROM customers
WHERE country = 'USA'
ORDER BY name;
```

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

customer_id	name	country
.	Alice Johnson	USA

Result Grid

```
92 -- 2. Aggregate with GROUP BY
93 • SELECT country, COUNT(*) AS num_customers
94 FROM customers
95 GROUP BY country
96 ORDER BY num_customers DESC;
```

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





	country	num_customers
▶	USA	1
	Canada	1
	UK	1

Result 2 x

```

111  -- 4. JOIN across three tables
112  • SELECT
113      CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
114      p.category,
115      SUM(oi.quantity * oi.price) AS total_spent
116  FROM order_items oi
117  JOIN orders o ON oi.order_id = o.order_id
118  JOIN products p ON oi.product_id = p.product_id
119  JOIN customers c ON o.customer_id = c.customer_id
120  GROUP BY c.customer_id, p.category
121  ORDER BY total_spent DESC
122  LIMIT 10;

```



Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	customer_name	category	total_spent
▶	Alice Johnson	Electronics	1799.96
	Bob Smith	Books	29.99

```

98  -- 3. JOIN: Find orders with customer info
99  • SELECT
100      o.order_id,
101      o.order_date,
102      CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
103      o.status
104  FROM orders o
105  INNER JOIN customers c ON o.customer_id = c.customer_id
106  WHERE o.order_date > '2023-01-01'
107  ORDER BY o.order_date DESC
108  LIMIT 10000;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	order_id	order_date	customer_name	status
▶	1003	2025-06-27	Alice Johnson	Delivered
	1002	2025-06-26	Bob Smith	Processing
	1001	2025-06-25	Alice Johnson	Shipped

```

125  -- 5. Subquery: Products above average price
126  • SELECT product_id, product_name, price
127  FROM products
128  WHERE price > (
129      SELECT AVG(price)
130      FROM products
131  )
132  LIMIT 10000;
133
134

```

Result Grid   Filter Rows:  Edit:    Export/Import:   Wrap Cell Content: 

	product_id	product_name	price
▶	101	Smartphone	699.99
*	NULL	NULL	NULL

```

135  -- 6. Subquery in SELECT: Customer total spend
136  • SELECT
137      c.customer_id,
138      CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
139      (
140          SELECT SUM(oi.quantity * oi.price)
141          FROM orders o
142          JOIN order_items oi ON o.order_id = oi.order_id
143          WHERE o.customer_id = c.customer_id
144      ) AS total_spent
145  FROM customers c
146  ORDER BY total_spent DESC
147  LIMIT 5;

```

Result Grid			
	customer_id	customer_name	total_spent
▶	1	Alice Johnson	1799.96
	2	Bob Smith	29.99
	3	Charlie Brown	NULL

**SCHEMAS**  
 Filter objects  
 e\_commerce  
 ecom  
 ecommerce\_db  
 Tables  
 customers  
 order\_items  
 orders  
 payments  
 products  
 Views  
 customer\_spending  
 customer\_id  
 customer\_name  
 total\_spent  
 Stored Procedures  
 Administration  
 Schemas  
 Information

Limit to 10000 rows  
 149 -- 7. Create VIEW  
 150  
 151 • CREATE VIEW customer\_spending AS  
 152 SELECT  
 153 c.customer\_id,  
 154 CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,  
 155 SUM(oi.quantity \* oi.price) AS total\_spent  
 156 FROM customers c  
 157 JOIN orders o ON c.customer\_id = o.customer\_id  
 158 JOIN order\_items oi ON o.order\_id = oi.order\_id  
 159 GROUP BY c.customer\_id;  
 160  
 161  
 Output  
 Action Output

```

162  -- 8. Index Optimization
163  • CREATE INDEX idx_orders_customer ON orders(customer_id);
164  -- Run EXPLAIN on a query to check use of index:
165  • EXPLAIN SELECT * FROM orders WHERE customer_id = 12345;
166

```



Result Grid											
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered
▶	1	SIMPLE	orders	NULL	ref	idx_orders_customer	idx_orders_customer	5	const	1	100.00

```

167  -- 9. Left Join and Right Join
168  • SELECT
169      c.customer_id,
170      CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
171      o.order_id,
172      o.order_date
173  FROM customers c
174  LEFT JOIN orders o ON c.customer_id = o.customer_id
175  ORDER BY c.customer_id;
176
177

```

```
177 -- 10. Left and Right Join
178
179 • SELECT
180     o.order_id,
181     o.order_date,
182     CONCAT(c.first_name, ' ', c.last_name) AS customer_name
183 FROM customers c
184 RIGHT JOIN orders o ON c.customer_id = o.customer_id
185 ORDER BY o.order_id;
186
187
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	order_id	order_date	customer_name
▶	1001	2025-06-25	Alice Johnson
	1002	2025-06-26	Bob Smith
	1003	2025-06-27	Alice Johnson