**User Authentication & Registration**

SELECT \* FROM users WHERE username = ?

SELECT \* FROM users WHERE username = ? AND password = ? AND user\_role = 'admin'

SELECT \* FROM users WHERE username = ? OR email = ?

INSERT INTO users (full\_name, username, email, password, user\_role) VALUES (?, ?, ?, ?, ?)

SELECT user\_id FROM users WHERE username=?

**Threats**   
SELECT COUNT(\*) AS total FROM threats

SELECT \* FROM threats WHERE submitted\_by = ?

SELECT \* FROM threats WHERE (threat\_name LIKE ? OR description LIKE ?) ORDER BY reported\_date DESC

SELECT \* FROM threats WHERE threat\_name LIKE ? OR affected\_industry LIKE ?

SELECT \* FROM threats WHERE id = ?

INSERT INTO threats (threat\_name, description, severity, affected\_industry, reported\_date, submitted\_by) VALUES (?, ?, ?, ?, ?, ?)

UPDATE threats SET threat\_name = ?, description = ?, severity = ?, affected\_industry = ? WHERE id = ?

DELETE FROM threats WHERE id = ?

SELECT t.\*, u.full\_name FROM threats t JOIN users u ON t.submitted\_by = u.username ORDER BY t.reported\_date DESC

SELECT \* FROM threats WHERE ... (with dynamic WHERE clause for filters)

**Alerts**

SELECT COUNT(\*) AS total FROM alerts

SELECT \* FROM alerts ORDER BY created\_at DESC

SELECT \* FROM alerts WHERE id = ?

INSERT INTO alerts (title, message, type, created\_at) VALUES (?, ?, ?, NOW())

UPDATE alerts SET title = ?, message = ?, type = ? WHERE id = ?

DELETE FROM alerts WHERE id = ?

SELECT title FROM alerts WHERE id = ?

**users**

SELECT COUNT(\*) AS total FROM users

SELECT user\_id FROM users WHERE username=?

**logs**

SELECT COUNT(\*) AS total FROM users

SELECT COUNT(\*) AS total FROM threats

SELECT COUNT(\*) AS total FROM alerts

SELECT \* FROM alerts ORDER BY created\_at DESC LIMIT 5

**fetching threat name for logging**

SELECT threat\_name FROM threats WHERE id = ?

**Filtering**   
  
-- 1. Filter by industry

SELECT \* FROM threats WHERE affected\_industry = 'Banking' ORDER BY reported\_date DESC LIMIT 10;

-- 2. Filter by severity

SELECT \* FROM threats WHERE severity = 'High' ORDER BY reported\_date DESC LIMIT 10;

-- 3. Filter by threat name (partial match)

SELECT \* FROM threats WHERE threat\_name LIKE '%ransomware%' ORDER BY reported\_date DESC LIMIT 10;

-- 4. Filter by submitted\_by (user name, partial match)

SELECT \* FROM threats WHERE submitted\_by LIKE '%john%' ORDER BY reported\_date DESC LIMIT 10;

-- 5. Filter by industry AND severity

SELECT \* FROM threats WHERE affected\_industry = 'Healthcare' AND severity = 'Critical' ORDER BY reported\_date DESC LIMIT 10;

-- 6. Filter by multiple fields (industry, severity, threat name, user name)

SELECT \* FROM threats

WHERE affected\_industry = 'Government'

AND severity = 'Medium'

AND threat\_name LIKE '%phish%'

AND submitted\_by LIKE '%alice%'

ORDER BY reported\_date DESC LIMIT 10;

**Stored Procedure**

DELIMITER //

CREATE PROCEDURE add\_threat\_with\_log (

IN p\_threat\_name VARCHAR(255),

IN p\_description TEXT,

IN p\_severity VARCHAR(50),

IN p\_industry VARCHAR(100),

IN p\_date DATE,

IN p\_submitter VARCHAR(100),

IN p\_user\_id INT

)

BEGIN

INSERT INTO threats (threat\_name, description, severity, affected\_industry, reported\_date, submitted\_by)

VALUES (p\_threat\_name, p\_description, p\_severity, p\_industry, p\_date, p\_submitter);

INSERT INTO logs (user\_id, action)

VALUES (p\_user\_id, CONCAT('Submitted new threat: ', p\_threat\_name));

END //

DELIMITER ;

CALL add\_threat\_with\_log('Phishing Attack', 'Tricks users into clicking fake links', 'Medium', 'Banking', CURDATE(), 'haris', 38);

**Trigger**

DELIMITER //

CREATE TRIGGER log\_alert\_insert

AFTER INSERT ON alerts

FOR EACH ROW

BEGIN

INSERT INTO logs (user\_id, action)

VALUES (28, CONCAT('Added new alert: ', NEW.title)); -- default admin user\_id = 28

END //

DELIMITER ;

**Function**

DELIMITER //

CREATE FUNCTION get\_threat\_count\_by\_severity(p\_severity VARCHAR(50))

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE threat\_count INT;

SELECT COUNT(\*) INTO threat\_count

FROM threats

WHERE severity = p\_severity;

RETURN threat\_count;

END //

DELIMITER ;