

```
CREATE TABLE departments (  
    department_id INT PRIMARY KEY AUTO_INCREMENT,  
    department_name VARCHAR(100) NOT NULL  
);
```

```
-----  
CREATE TABLE suppliers (  
    supplier_id INT PRIMARY KEY AUTO_INCREMENT,  
    supplier_name VARCHAR(100) NOT NULL,  
    contact_email VARCHAR(100) NOT NULL  
);
```

```
-----  
CREATE TABLE items (  
    item_id INT PRIMARY KEY AUTO_INCREMENT,  
    item_name VARCHAR(100) NOT NULL,  
    unit VARCHAR(20) NOT NULL,  
    unit_price DECIMAL(10,2) NOT NULL CHECK (unit_price > 0)  
);
```

```
-----  
CREATE TABLE inventory (  
    item_id INT PRIMARY KEY,  
    quantity_available INT NOT NULL CHECK (quantity_available >= 0),  
    last_updated DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
    FOREIGN KEY (item_id) REFERENCES items(item_id)  
);
```

```
-----  
CREATE TABLE procurement_requests (  
    request_id INT PRIMARY KEY AUTO_INCREMENT,  
    department_id INT NOT NULL,  
    request_date DATE NOT NULL,  
    status ENUM('Pending','Approved','Delivered') NOT NULL,  
    FOREIGN KEY (department_id) REFERENCES departments(department_id)  
);
```

```
-----  
CREATE TABLE procurement_details (  
    detail_id INT PRIMARY KEY AUTO_INCREMENT,  
    request_id INT NOT NULL,  
    item_id INT NOT NULL,  
    supplier_id INT NOT NULL,  
    quantity INT NOT NULL CHECK (quantity > 0),  
  
    FOREIGN KEY (request_id) REFERENCES procurement_requests(request_id),  
    FOREIGN KEY (item_id) REFERENCES items(item_id),
```

```
FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id)
);
```

```
-----
DELIMITER $$
```

```
CREATE PROCEDURE submit_procurement (
    IN p_department_id INT,
    IN p_request_date DATE,
    IN p_item_id INT,
    IN p_supplier_id INT,
    IN p_quantity INT
)
BEGIN
    DECLARE req_id INT;

    START TRANSACTION;

    INSERT INTO procurement_requests
        (department_id, request_date, status)
    VALUES
        (p_department_id, p_request_date, 'Pending');

    SET req_id = LAST_INSERT_ID();

    INSERT INTO procurement_details
        (request_id, item_id, supplier_id, quantity)
    VALUES
        (req_id, p_item_id, p_supplier_id, p_quantity);

    COMMIT;
END$$
```

```
DELIMITER ;
```

```
(TRIGGER STORED PROCEDURE ACID)
```

```
-----
DELIMITER $$
```

```
CREATE TRIGGER trg_update_inventory
AFTER UPDATE ON procurement_requests
FOR EACH ROW
BEGIN
    IF NEW.status = 'Delivered' AND OLD.status <> 'Delivered' THEN
```

```
INSERT INTO inventory (item_id, quantity_available)
SELECT item_id, quantity
FROM procurement_details
WHERE request_id = NEW.request_id
ON DUPLICATE KEY UPDATE
    quantity_available = quantity_available + VALUES(quantity_available);
END IF;
END$$
```

DELIMITER ;

(INVENTORY UPDATE)

-----

DELIMITER \$\$

```
CREATE TRIGGER trg_prevent_skip_approval
BEFORE UPDATE ON procurement_requests
FOR EACH ROW
BEGIN
    IF NEW.status = 'Delivered' AND OLD.status <> 'Approved' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: You cannot deliver a request that is not approved.';
    END IF;
END$$
```

DELIMITER ;

-----

\*VIEWS\*

```
CREATE VIEW vw_pending_requests AS
SELECT
    pr.request_id,
    d.department_name,
    pr.request_date
FROM procurement_requests pr
JOIN departments d ON pr.department_id = d.department_id
```

```
WHERE pr.status = 'Pending';
```

```
CREATE VIEW vw_item_procurement_summary AS  
SELECT
```

```
    i.item_name,  
    SUM(pd.quantity) AS total_quantity
```

```
FROM items i
```

```
JOIN procurement_details pd ON i.item_id = pd.item_id
```

```
GROUP BY i.item_name;
```

```
CREATE VIEW vw_supplier_transactions AS
```

```
SELECT
```

```
    s.supplier_name,  
    COUNT(pd.detail_id) AS total_transactions
```

```
FROM suppliers s
```

```
JOIN procurement_details pd ON s.supplier_id = pd.supplier_id
```

```
GROUP BY s.supplier_name;
```

```
(DEPARTMENTS)
```

```
INSERT INTO departments (department_name)
```

```
VALUES
```

```
('Computer Laboratory'),
```

```
('IT Department'),
```

```
('Business Administration Department'),
```

```
('Education Department'),
```

```
('Hospitality Management Department'),
```

```
('Tourism Management Department'),
```

```
('Psychology Department'),
```

```
('Criminology Department'),
```

```
('Engineering Department'),
```

```
('Accountancy Department');
```

```
-----
```

```
(SUPPLIERS)
```

```
INSERT INTO suppliers (supplier_name, contact_email)
```

```
VALUES
```

```
('Tech Supplies Inc.', 'techsupplies@email.com'),
```

```
('Digital World PH', 'digitalworld@email.com'),
```

```
('EduPrint Stationery', 'contact@eduprintph.com'),
```

```
('SchoolHub Trading', 'info@schoolhubtrading.com'),
```

```
('Metro Office Depot', 'support@metroofficedepot.ph'),
```

```
('PrimeTech Solutions', 'sales@primetech.ph'),
```

```
('BrightStar Paper & Supplies', 'orders@brightstarpaper.com'),
```

('Innovative Learning Tools Co.', 'service@iltc.ph'),  
('Urban Office & IT Supply', 'hello@urbanoffice.ph'),  
('NorthGate Electronics', 'sales@northgateelec.com');

-----  
(ITEMS)

INSERT INTO items (item\_name, unit, unit\_price)

VALUES

('Keyboard', 'pcs', 450.00),  
('Mouse', 'pcs', 250.00),  
('Monitor', 'unit', 6500.00),  
('Ballpen Black', 'pcs', 12.00),  
('Ballpen Blue', 'pcs', 12.00),  
('Pencil HB No.2', 'pcs', 8.00),  
('Notebook 80 pages', 'pcs', 25.00),  
('Envelope Long Brown', 'pcs', 3.00),  
('Permanent Marker Black', 'pcs', 35.00),  
('Whiteboard Marker Black', 'pcs', 30.00),  
('Cartolina White', 'pcs', 8.00),  
('Cartolina Colored', 'pcs', 10.00),  
('Folder Long Manila', 'pcs', 4.00),  
('Scissors Medium', 'pcs', 45.00),  
('Stapler Standard', 'pcs', 90.00),  
('Staple Wires (No. 10)', 'box', 25.00),  
('Ruler 12 inches', 'pcs', 15.00),  
('Clipboard Long', 'pcs', 35.00),  
('Correction Tape', 'pcs', 20.00),  
('Glue Stick Small', 'pcs', 18.00),  
('Highlighter Yellow', 'pcs', 25.00),  
('Expanding Envelope Long', 'pcs', 15.00);

-----  
UPDATE procurement\_requests

SET status = 'Approved'

WHERE request\_id = 1;

(APPROVED)

UPDATE procurement\_requests

SET status = 'Delivered'

WHERE request\_id = 1;

(DELIVERED)

(ADD INVENTORY DATA)