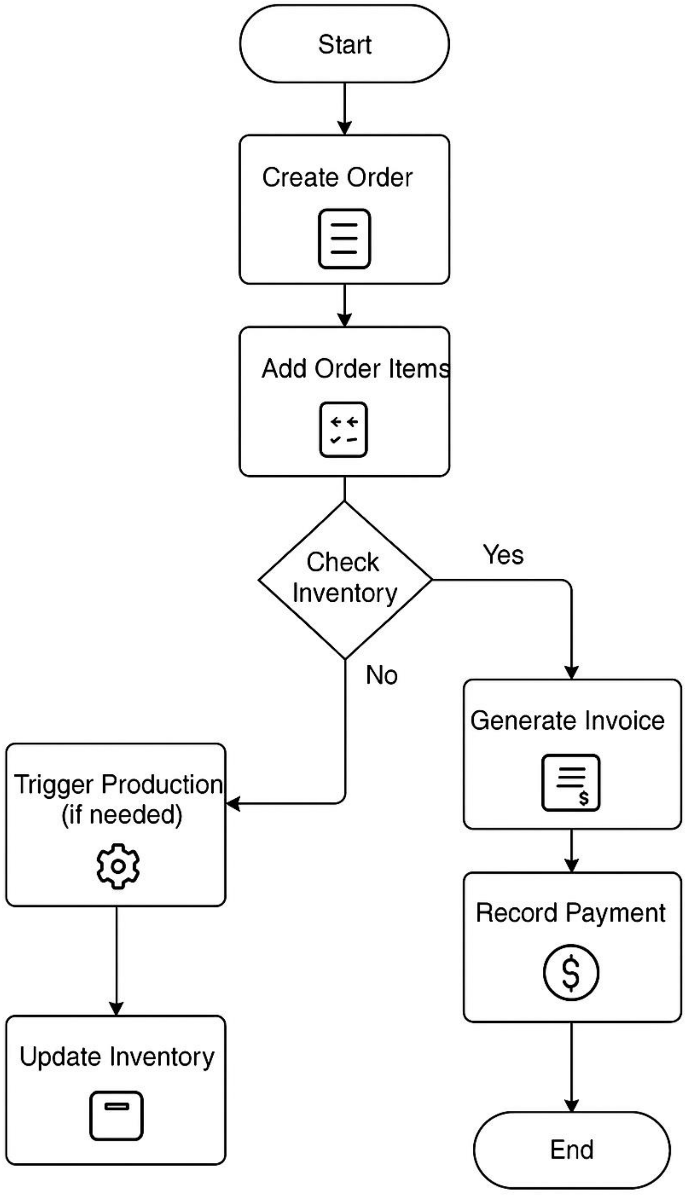
Order Kraft-Invoicing & Payments

Order to Invoice Generation Flow:

1. **User (Procurement Officer)** places an **Order**
   - Enters order details like date, status, and total amount
   - Assigned to a procurement_officer_id
2. **Order Items** are added
   - Each item includes quantity and unit price
   - Linked to the main order via order_id
3. **Inventory Check**
   - System checks if products exist in **Inventory**
   - If **stock is insufficient**, trigger **Production Schedule**
4. **Production Process** (if needed)
   - Initiate **Production Schedule**
   - Break into **Production Tasks**
   - Use **Production Units** and **Raw Materials**
5. **Inventory Update**
   - Once production is complete or items are confirmed in stock
   - Inventory is updated (via inventory_id, product_id)
6. **Generate Invoice**
   - Linked to order_id
   - Contains invoice date, due date, total amount
7. **Payment Process**
   - Payment created and linked to order
   - Includes payment method, date, and status

Flowchart:

Order to Invoice Generation Flow

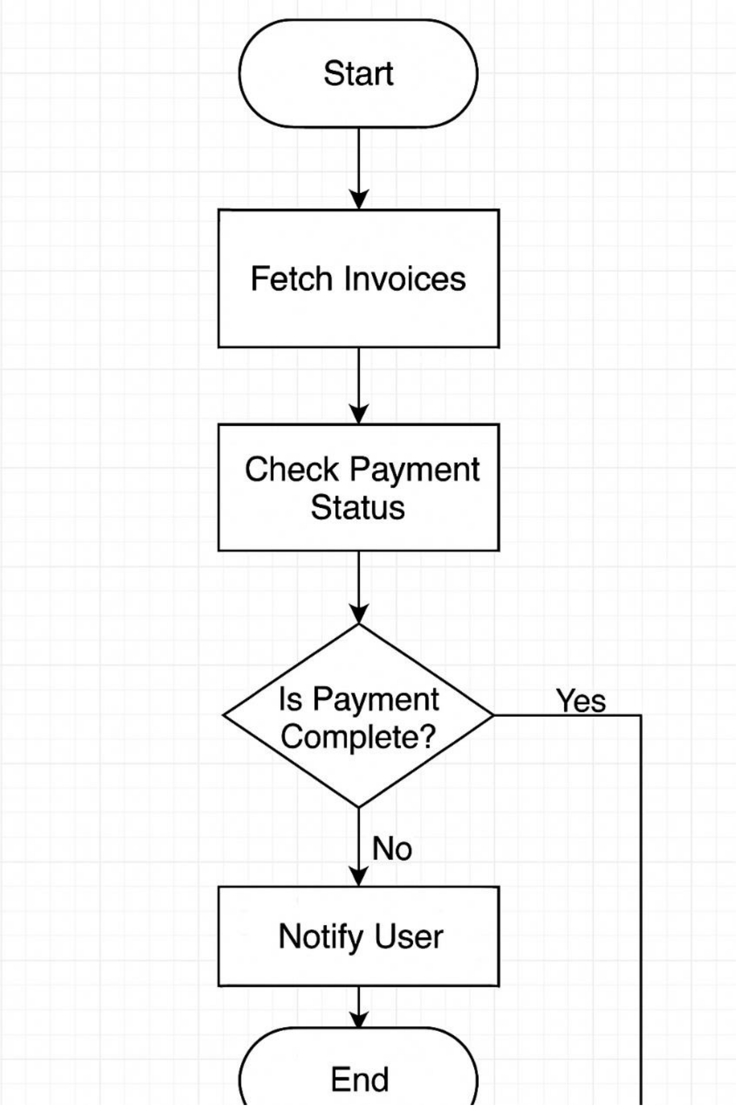


Invoice Due Notification Flow:

1. **Start**: System Scheduler or Cron Job Initiates Daily Check
2. **Fetch Invoices**
   - Retrieve all records from the invoice table
   - Check if due_date is approaching (e.g., within 3 days) or has passed
3. **Check Payment Status**
   - Match each invoice against entries in the payment table (by order_id)
   - Determine if full payment has been made
4. **Decision Point**: Is Payment Complete?
   - **Yes**: No action needed
   - **No**: Continue to notification step
5. **Notify User**
   - Identify user_id via order → procurement_officer_id → user
   - Send Email/Alert Notification about the invoice due
6. **Log Notification (Optional)**
   - Optionally store notification history (not shown in schema but assumed)

7.End.

Flow Chart:


Start
Fetch Invoices
Check Payment Status
Is Payment Complete?
Yes
No
Notify User
End

Payment Confirmation & Reconciliation Flow:

1.Start

(Initiated by user payment or scheduled reconciliation process)

2. **Fetch Payments**

- Retrieve data from the payment table

3. Match Payment to Order

- Use order_id to link payment to a specific order

4. Validate Payment Details

- Check amount, payment_date, and status fields
- Optionally verify against external bank/payment gateway

5. Reconcile with Invoice

- Compare payment amount with invoice.total_amount
- Ensure status matches (Paid, Partial, or Pending)

6. **Decision Point**: Is Full Payment Received?

- **Yes**: Mark Invoice as **Paid**
- **No**: Update status to **Partial** or **Pending**

7. Update Order & Invoice Status

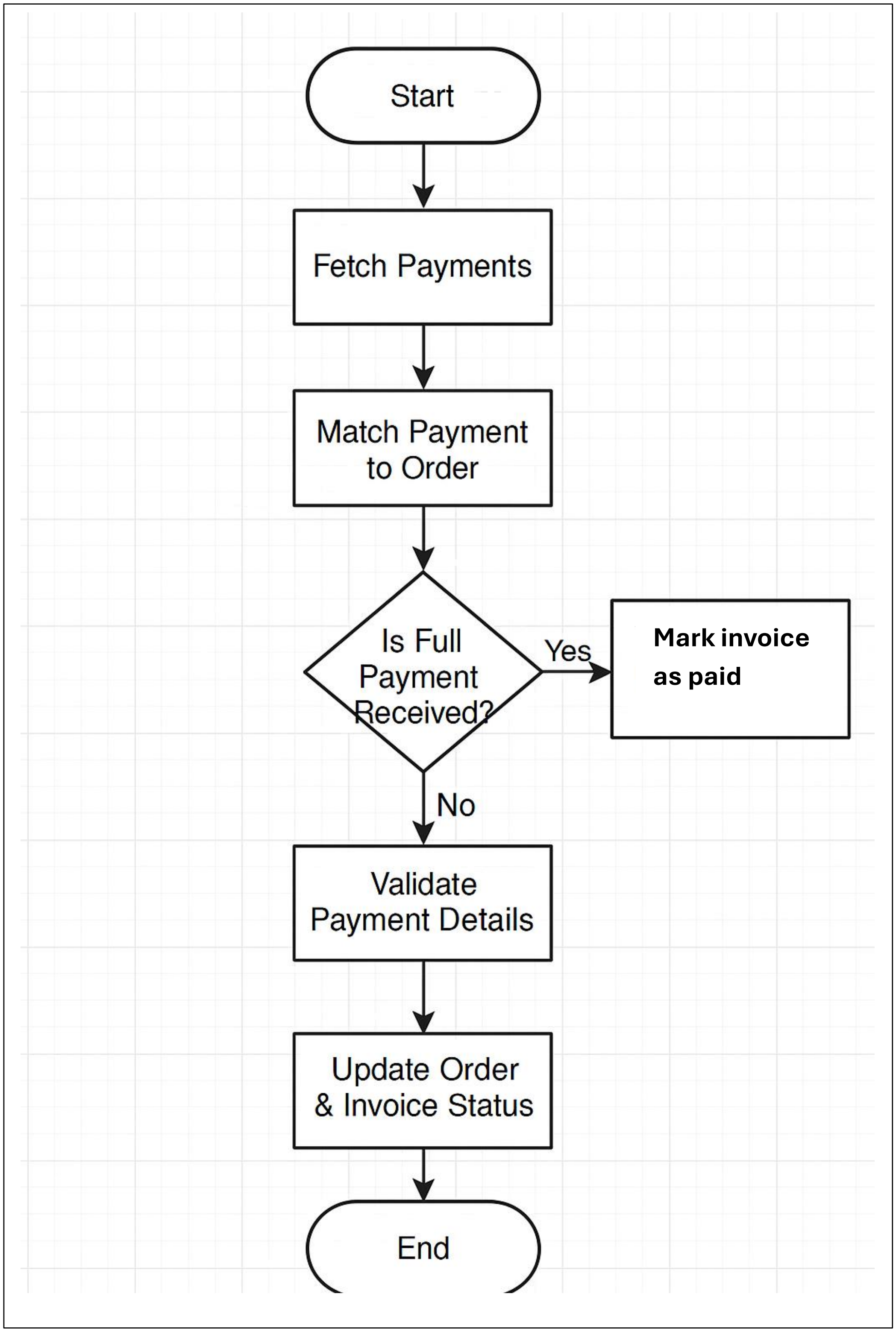
- Mark order as Completed or Awaiting Payment
- Update invoice status or tracking field (if implemented)

8. Log Audit Trail

- Record reconciliation result in logs (not shown in schema, assumed optional)

9.End

Flow Chart:


Start
Fetch Payments
Match Payment to Order
Is Full Payment Received?
Yes
Mark invoice as paid
No
Validate Payment Details
Update Order & Invoice Status
End