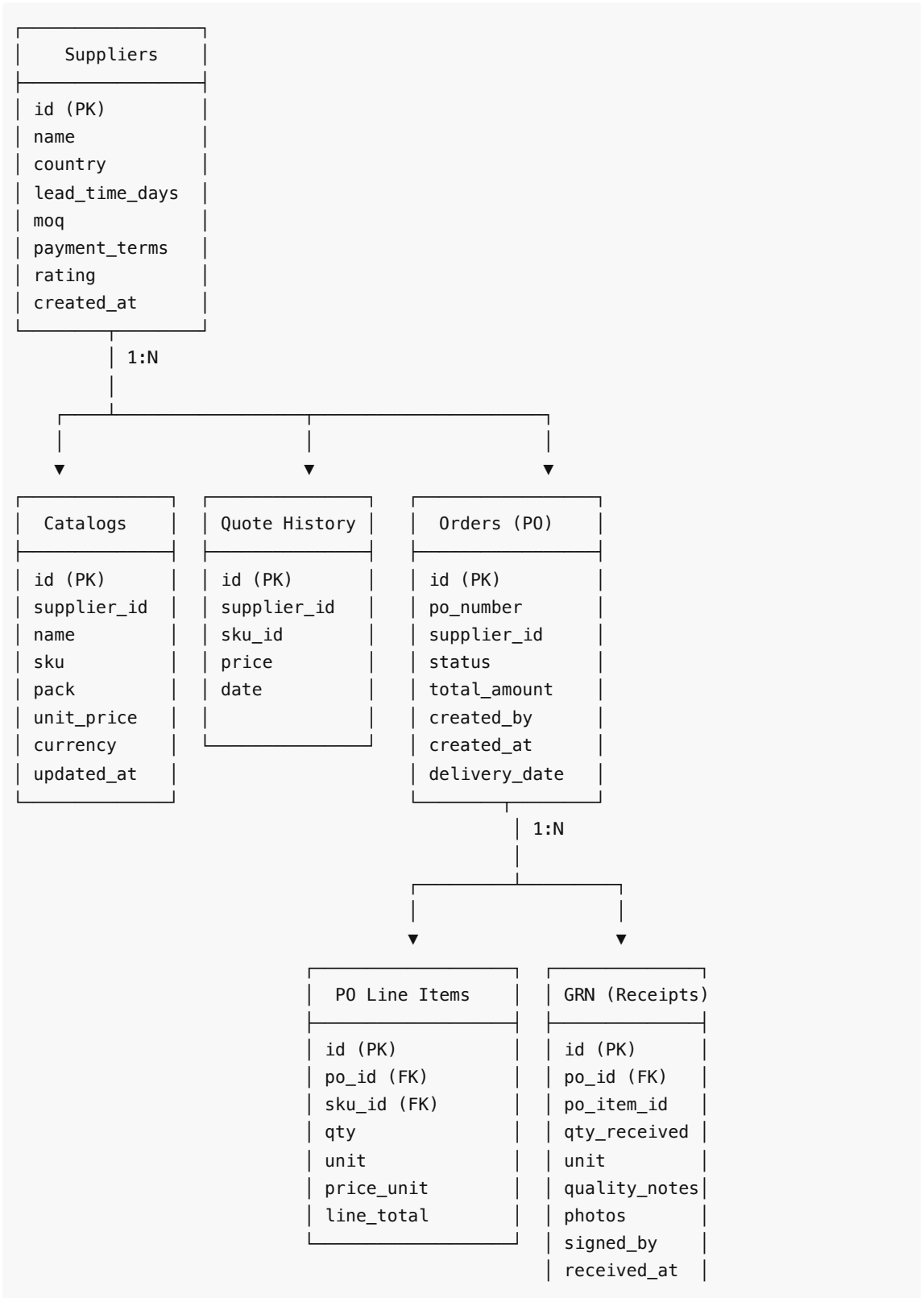
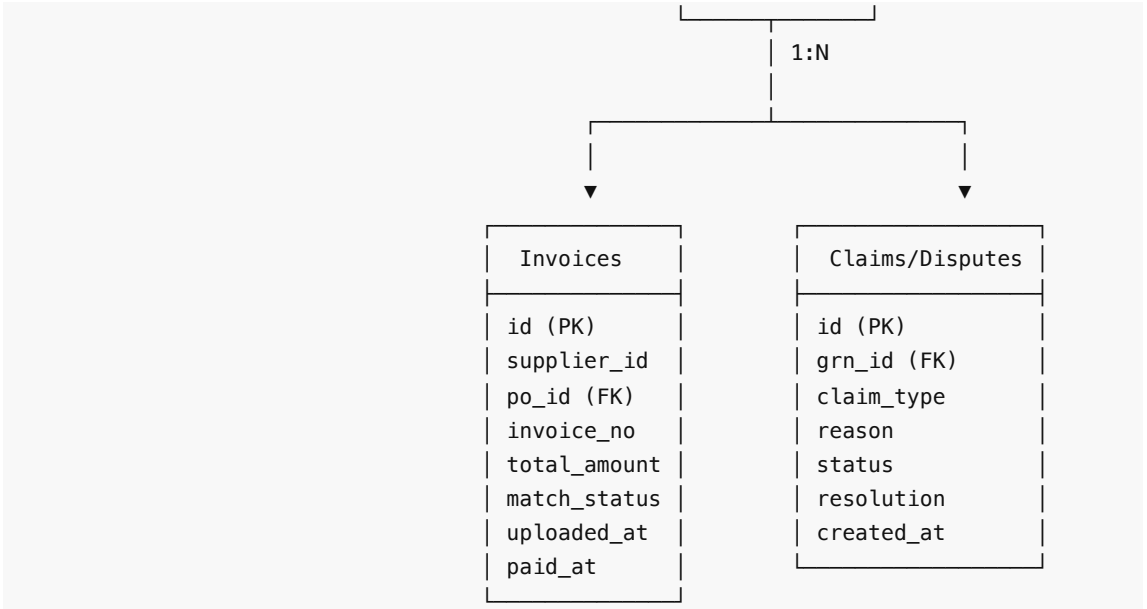
# Database Schema & Data Model

## Entity Relationship Diagram






```
┌──────────────────┐
│ Normalized SKUs  │
├──────────────────┤
│ id (PK)          │
│ supplier_id      │
│ sku              │
│ name             │
│ normalized_name  │
│ pack_info        │ {count, size, unit}
│ price_per_kg     │
│ equiv_group_id   │ ← links to equivalents
│ created_at       │
└──────────────────┘

┌──────────────────┐
│  Inventory       │
├──────────────────┤
│ id (PK)          │
│ normalized_sku_id│
│ location         │
│ qty_on_hand      │
│ par_level        │
│ unit             │
│ last_check       │
│ expiry_batch     │
└──────────────────┘

┌──────────────────┐
│  Audit Logs      │
├──────────────────┤
│ id (PK)          │
│ agent_name       │
│ action           │
```

```
| tool_name        |
| tool_input       |
| tool_output      |
| status           |
| error (opt)      |
| user_approval    |
| created_at       |
└──────────────────┘
```

---

## Core Data Models (Pydantic + SQL)

### Supplier

```
Supplier {
  id: UUID (Primary Key)
  name: str
  country: str
  lead_time_days: int
  moq: float (minimum order qty)
  payment_terms: str ("net 30", "prepay", etc.)
  rating: float (0–5, from historical reliability)
  is_active: bool
  created_at: datetime
  updated_at: datetime
}
```

### Catalog / SKU

```
CatalogSKU {
  id: UUID (Primary Key)
  supplier_id: UUID (Foreign Key)
  sku: str (supplier's SKU code)
  name: str (supplier's product name)
  pack: str ("10 x 1kg", "5lb", "12/500g")
  unit_price: float (supplier's price)
  currency: str ("USD", "AED", etc.)
  created_at: datetime
  updated_at: datetime
}
```

### Normalized SKU

```
NormalizedSKU {
  id: UUID (Primary Key)
  catalog_sku_id: UUID (Foreign Key)
  normalized_name: str (parsed & cleaned)
  pack_info: JSON {
    count: int
    size: float
```

```
    unit: str ("kilogram", "gram", "pound", "ounce")
  }
  total_weight_kg: float
  price_per_kg: float
  equiv_group_id: UUID (for "apples == pommes")
  created_at: datetime
}
```

### Purchase Order (PO)

```
PurchaseOrder {
  id: UUID (Primary Key)
  po_number: str (unique, auto-incremented)
  supplier_id: UUID (Foreign Key)
  status: str ("draft", "approved", "sent", "confirmed", "delivered", "invoiced",
"paid")
  branch_id: UUID (which restaurant branch)
  created_by: UUID (user who created/approved)
  created_at: datetime
  delivery_date: date (promised delivery)
  total_amount: float (in supplier's currency)
  currency: str
  approved_at: datetime (when manager approved)
  line_items: List[POLineItem]
}

POLineItem {
  id: UUID (Primary Key)
  po_id: UUID (Foreign Key)
  normalized_sku_id: UUID (Foreign Key)
  qty: float
  unit: str ("kg", "lb", "unit")
  price_per_unit: float
  line_total: float
}
```

### GRN (Goods Received Note)

```
GRN {
  id: UUID (Primary Key)
  po_id: UUID (Foreign Key)
  received_by: str (receiver name)
  received_at: datetime
  status: str ("accepted", "partial", "rejected", "disputed")
  notes: str
  photos: List[str] (URLs to proof images)
  signature: str (base64 or URL)
  line_items: List[GRNLineItem]
}

GRNLineItem {
```

```
  id: UUID (Primary Key)
  grn_id: UUID (Foreign Key)
  po_item_id: UUID (Foreign Key)
  qty_received: float
  unit: str
  quality_notes: str ("damaged", "expired", "correct", etc.)
  substitution: bool (was substitution accepted?)
}
```

### Invoice

```
Invoice {
  id: UUID (Primary Key)
  supplier_id: UUID (Foreign Key)
  po_id: UUID (Foreign Key)
  grn_id: UUID (Foreign Key, optional)
  invoice_number: str (supplier's invoice no)
  invoice_date: date
  total_amount: float
  currency: str
  uploaded_at: datetime
  ocr_extracted: bool (was OCR successful?)
  match_status: str ("unmatched", "2way_match", "3way_match", "exception")
  exception_reason: str (if status="exception")
  line_items: List[InvoiceLineItem]
  approval_by: UUID (finance person)
  approved_at: datetime
}

InvoiceLineItem {
  id: UUID (Primary Key)
  invoice_id: UUID (Foreign Key)
  description: str (invoice line)
  qty_billed: float
  unit_price: float
  line_total: float
}
```

### Inventory

```
InventorySnapshot {
  id: UUID (Primary Key)
  normalized_sku_id: UUID (Foreign Key)
  branch_id: UUID (location)
  location_type: str ("dry", "chiller", "freezer")
  qty_on_hand: float
  unit: str
  par_level: float (target qty)
  reorder_point: float (trigger qty)
  expiry_date: date (oldest batch)
  batch_number: str
```

```
  last_checked: datetime
  updated_at: datetime
}
```

### Audit Log (All Agent Actions)

```
AuditLog {
  id: UUID (Primary Key)
  timestamp: datetime
  agent_name: str ("catalog_agent", "purchasing_agent", etc.)
  action: str ("suggest_cart", "compare_quotes", "match_invoice", etc.)
  tool_name: str (which tool was called)
  tool_input: JSON (what was passed)
  tool_output: JSON (what came back)
  error: str (if any)
  user_action: str ("approved", "edited", "rejected", etc.)
  created_by: UUID (user who took action)
  notes: str
}
```

### Suggested Cart (in-memory during draft)

```
SuggestedCart {
  id: UUID (Primary Key, optional for draft)
  branch_id: UUID
  created_by_agent: str ("purchasing_agent")
  status: str ("draft", "pending_approval", "approved", "rejected")
  items: List[SuggestedCartItem]
  reasoning: JSON (explainability)
  created_at: datetime
  expires_at: datetime (cart is stale after 24h)
}

SuggestedCartItem {
  id: UUID
  normalized_sku_id: UUID
  qty: float
  unit: str
  supplier_id: UUID
  suggested_price_per_unit: float
  reasoning: str (why this qty & supplier?)
  can_substitute: bool
  substitutes: List[UUID] (alternative SKU IDs)
}
```

---

## State Transitions

### PO Lifecycle

```
draft → approved (manager signs off)
     ↓
   sent (transmitted to supplier)
     ↓
 confirmed (supplier confirms)
     ↓
 dispatched (in transit)
     ↓
delivered (GRN recorded)
     ↓
invoiced (invoice received & matched)
     ↓
paid (payment processed)
```

### Invoice Reconciliation States

```
unmatched → 2way_match (PO=Invoice qty/price)
         ↓
      accepted → paid

         OR

      exception (mismatch) → 3way_match (check GRN)
                           ↓
                      resolved (claim settled)
                      ↓
                      credited/adjusted/paid
```

### Low Stock Trigger → AI Cart → Approval → PO

```
Inventory.qty < par → Alert → AI Draft Cart
                            ↓
                     Manager Reviews
                            ↓
                     ✓ Approve → PO Created
                     ✗ Edit    → Modify & Resubmit
                     ✗ Reject  → Manual later
```