Databricks Infrastructure Requirements - Deal Desk Project

To: Anthony Perez, VP Innovation & Growth Programs
From: Ethan Sam, Growth & Innovation Associate
Re: Database Infrastructure for Trading Department Deal Desk Solution
Date: August 2025
Timeline: 6-8 weeks total development

📋 Executive Summary

PROJECT STATUS DASHBOARD

Aspect	Details
Business Sponsor	Van Ngo (RVP Trading, Northeast)
Technical Lead	Ethan Sam (Growth & Innovation)
Infrastructure Lead	Anthony Perez (VP Innovation & Growth)
Development Timeline	6-8 weeks total development
Current Status	⚠️ **Blocked** - Awaiting Database Setup
Risk Level	🔴 **HIGH** - Data loss prevents production launch

The Challenge: Van's Trading Department needs a Deal Desk application to streamline commercial deal approvals, but we're blocked by data persistence requirements. Current in-memory storage loses all data on server restart.

The Ask: Databricks infrastructure setup to enable persistent data storage for production deployment.

🎯 Project Team & Organizational Structure

Team Hierarchy & Collaboration Model

Innovation & Growth Programs Department *(Anthony Perez, VP)* - **Ethan Sam** *(Growth & Innovation Associate)* - Technical implementation & database migration - Schema deployment and application integration - Migration testing and performance validation

Trading Department *(Cross-departmental collaboration)* - **Van Ngo** *(RVP Trading, Northeast)* - Business requirements definition - User acceptance testing with trading teams - Production rollout and change management

Team Responsibilities Matrix

Team Member	Department	Primary Role	Key Deliverables
Anthony Perez	Innovation & Growth	Strategic Infrastructure Lead	Databricks provisioning, architecture decisions, resource allocation
Ethan Sam	Innovation & Growth	Technical Implementation Lead	Database schema deployment, application migration, testing coordination
Van Ngo	Trading	Business Owner & End-User Champion	Requirements validation, user acceptance testing, trading team rollout

---

📊 Current State Analysis

Technical Problem

Technical Architecture Transformation

Current State: High-Risk In-Memory Storage

```
React Frontend → Express Server → In-Memory Storage (MemStorage)
                                          ↓
                                  💥 CRITICAL ISSUE:
                                  Data lost on every restart
                                  No persistence or recovery
```

Target State: Enterprise-Grade Persistent Storage

```
React Frontend → Express Server → Databricks Persistent Database
                                          ↓
                                  ✅ SOLUTION BENEFITS:
                                  • Zero data loss guarantee
                                  • Full audit trail capability
                                  • Scalable concurrent access
                                  • Enterprise backup & recovery
```

Business Impact Metrics

Current System Configuration *(Source: businessConstants.ts lines 24-29)*

Approval Level	Dollar Threshold	Business Impact
Manager	$50,000	Department-level deals
Director	$100,000	Multi-department coordination
VP	$500,000	Strategic business decisions
SVP	$1,000,000	Executive-level approvals

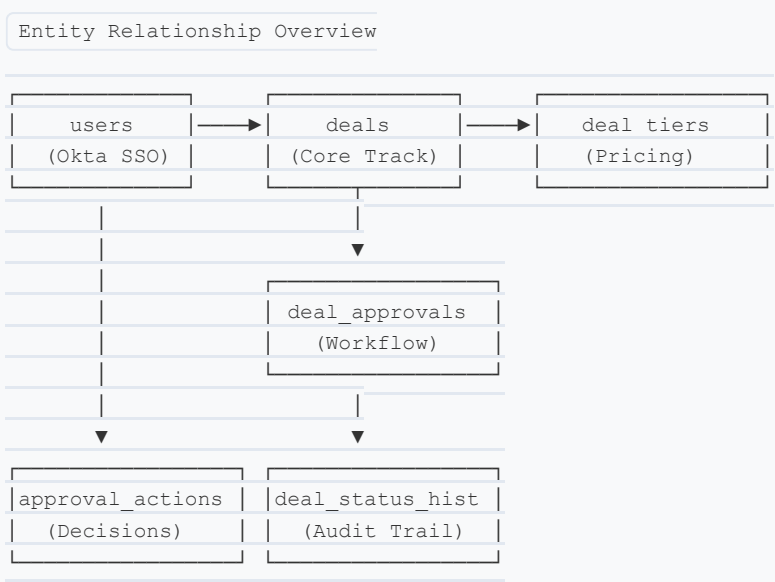
Trading Department Impact: - **Current State:** Manual approval processes, no pipeline visibility - **Risk:** Data loss prevents tracking deals through these approval tiers - **Opportunity:** Streamlined workflow for Van's Northeast Trading team

---

🔄 Technical Requirements

Database Schema Architecture

Core Entity Relationships



Detailed Table Specifications

Source: `/shared/schema.ts` (Drizzle ORM definitions)

1. Core Tables

`users` - Authentication & Authorization

```sql
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  username TEXT NOT NULL UNIQUE,
  password TEXT NOT NULL,  -- Okta integration
  email TEXT NOT NULL UNIQUE,
  role TEXT NOT NULL DEFAULT 'seller',
    -- ENUM: seller, department_reviewer, approver, admin
  department TEXT,
    -- ENUM: trading, finance, creative, marketing, product, solutions, legal
  first name TEXT,
  last name TEXT,
  is_active BOOLEAN NOT NULL DEFAULT true,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);
```

`deals` - Core Deal Tracking (30+ fields)

```sql
CREATE TABLE deals (
  id SERIAL PRIMARY KEY,

  -- Deal Identification
  deal_name TEXT NOT NULL,
  reference_number TEXT NOT NULL UNIQUE,  -- Format: DEAL-YYYY-XXX
  email TEXT,

  -- Business Classification
  deal_type TEXT NOT NULL,          -- ENUM: grow, protect, custom
  sales_channel TEXT NOT NULL,      -- ENUM: holding_company, independent_agency, client_direct
  deal structure TEXT NOT NULL,     -- ENUM: tiered, flat commit
  region TEXT,                      -- ENUM: northeast, midwest, midatlantic, west, south

  -- Client Information
  advertiser_name TEXT,
  agency_name TEXT,

  -- Business Context
  business_summary TEXT,
  growth_opportunity_miq TEXT,
  growth opportunity_client TEXT,
  client_asks TEXT,
  growth_ambition DOUBLE PRECISION,

  -- Contract Terms
  term start date TEXT,             -- ISO 8601 format
  term_end_date TEXT,               -- ISO 8601 format
  contract_term INTEGER,            -- Calculated months

  -- Financial History
  previous year revenue DOUBLE PRECISION DEFAULT 0,
  previous_year_margin DOUBLE PRECISION DEFAULT 0,

  -- Status Management
  status TEXT NOT NULL DEFAULT 'submitted',
    -- ENUM: draft, scoping, converted, submitted, under review,
    --       negotiating, approved, contract_drafting, client_review, signed, lost

  -- Draft & Revision Management
  draft type TEXT,                  -- ENUM: scoping_draft, submission_draft
  revision_count INTEGER NOT NULL DEFAULT 0,
```

```
  is_revision BOOLEAN NOT NULL DEFAULT false,
  parent_submission_id INTEGER,    -- Self-reference for revisions
  revision_reason TEXT,
  last_revised_at TIMESTAMP,
  can_edit BOOLEAN NOT NULL DEFAULT true,
  draft_expires_at TIMESTAMP,

  -- Workflow Intelligence
  last_status_change TIMESTAMP DEFAULT NOW(),
  priority TEXT NOT NULL DEFAULT 'medium',  -- ENUM: critical, high, medium, low
  flow_intelligence TEXT,          -- ENUM: on_track, needs_attention

  -- System Fields
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);
```

`deal_tiers` - Tiered Pricing Structures

```
CREATE TABLE deal_tiers (
  id SERIAL PRIMARY KEY,
  deal_id INTEGER NOT NULL,                    -- FK to deals.id
  tier_number INTEGER NOT NULL,                -- 1, 2, 3, 4, 5 (max tiers: 5)
  annual_revenue DOUBLE PRECISION NOT NULL,
  annual_gross_margin DOUBLE PRECISION NOT NULL,  -- Decimal: 0.355 = 35.5%

  -- Incentive Structure
  category_name TEXT NOT NULL,                 -- Display: "Financial", "Resources"
  sub_category_name TEXT NOT NULL,             -- Display: "Discounts", "Bonuses"
  incentive_option TEXT NOT NULL,              -- "Volume Discount", "Growth Bonus"
  incentive_value DOUBLE PRECISION NOT NULL,   -- USD amount
  incentive_notes TEXT,

  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);
```

2. Approval Workflow Tables

`deal_approvals` - Multi-Stage Approval System

```sql
CREATE TABLE deal_approvals (
  id SERIAL PRIMARY KEY,
  deal_id INTEGER NOT NULL,                    -- FK to deals.id

  -- Approval Configuration
  approval_stage INTEGER NOT NULL,             -- 1: Dept Review, 2: Business Approval
  department TEXT NOT NULL,                    -- ENUM: trading, finance, creative, etc.
  required_role TEXT NOT NULL,                 -- Role needed for approval

  -- Status & Assignment
  status TEXT DEFAULT 'pending',               -- ENUM: pending, revision_requested, approved
  priority TEXT DEFAULT 'normal',              -- ENUM: normal, high, urgent
  assigned_to INTEGER,                         -- FK to users.id
  due_date TIMESTAMP NOT NULL,

  -- Review Details
  comments TEXT,
  revision_reason TEXT,                        -- When status = revision_requested
  reviewer_notes TEXT,
  completed_at TIMESTAMP,
  created_at TIMESTAMP DEFAULT NOW()
);
```

`approval_actions` - Individual Approval Decisions

```sql
CREATE TABLE approval_actions (
  id SERIAL PRIMARY KEY,
  approval_id INTEGER NOT NULL,                -- FK to deal_approvals.id

  action_type TEXT NOT NULL,                   -- ENUM: approve, reject, request_revision,
                                               --       comment, initiate, assign
  performed_by INTEGER NOT NULL,               -- FK to users.id
  comments TEXT,
  created_at TIMESTAMP DEFAULT NOW()
);
```

`deal_status_history` - Complete Audit Trail

```sql
CREATE TABLE deal_status_history (
  id SERIAL PRIMARY KEY,
  deal id INTEGER NOT NULL,                    -- FK to deals.id
  status TEXT NOT NULL,                        -- Current status
  previous_status TEXT,                        -- Previous status
  performed_by INTEGER,                        -- FK to users.id
  comments TEXT,
  changed_at TIMESTAMP DEFAULT NOW()
);
```

3. Supporting Tables

`advertisers` - Client Company Data

```sql
CREATE TABLE advertisers (
  id SERIAL PRIMARY KEY,
  name TEXT NOT NULL UNIQUE,
  previous_year_revenue DOUBLE PRECISION DEFAULT 0,
  previous year margin DOUBLE PRECISION DEFAULT 0,    -- Decimal format
  previous year profit DOUBLE PRECISION DEFAULT 0,
  previous_year_incentive_cost DOUBLE PRECISION DEFAULT 0,
  previous_year_client_value DOUBLE PRECISION DEFAULT 0,
  region TEXT,
  created at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);
```

`agencies` - Partner Data

```sql
CREATE TABLE agencies (
  id SERIAL PRIMARY KEY,
  name TEXT NOT NULL UNIQUE,
  type TEXT NOT NULL DEFAULT 'independent',  -- ENUM: holding_company, independent
  previous_year_revenue DOUBLE PRECISION DEFAULT 0,
  previous_year_margin DOUBLE PRECISION DEFAULT 0,
  previous year profit DOUBLE PRECISION DEFAULT 0,
  previous year incentive cost DOUBLE PRECISION DEFAULT 0,
  previous_year_client_value DOUBLE PRECISION DEFAULT 0,
  region TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);
```

Foreign Key Relationships & Constraints

```sql
-- Primary Relationships
ALTER TABLE deals ADD CONSTRAINT fk_deals_parent
  FOREIGN KEY (parent_submission_id) REFERENCES deals(id);

ALTER TABLE deal tiers ADD CONSTRAINT fk tiers deal
  FOREIGN KEY (deal_id) REFERENCES deals(id) ON DELETE CASCADE;

ALTER TABLE deal_approvals ADD CONSTRAINT fk_approvals_deal
  FOREIGN KEY (deal_id) REFERENCES deals(id) ON DELETE CASCADE;

ALTER TABLE deal_approvals ADD CONSTRAINT fk_approvals_assignee
  FOREIGN KEY (assigned_to) REFERENCES users(id);

ALTER TABLE approval actions ADD CONSTRAINT fk actions approval
  FOREIGN KEY (approval_id) REFERENCES deal_approvals(id) ON DELETE CASCADE;

ALTER TABLE approval_actions ADD CONSTRAINT fk_actions_user
  FOREIGN KEY (performed_by) REFERENCES users(id);

ALTER TABLE deal status history ADD CONSTRAINT fk history deal
  FOREIGN KEY (deal_id) REFERENCES deals(id) ON DELETE CASCADE;

ALTER TABLE deal_status_history ADD CONSTRAINT fk_history_user
  FOREIGN KEY (performed_by) REFERENCES users(id);
```

Business Rules & Validation

From `/shared/schema.ts` validation rules:

- **Growth Ambition:** Minimum $1M (z.number().min(1000000))
- **Tier Numbers:** 1-5 only (max 5 tiers per deal)
- **Gross Margin:** 0-1 decimal range (35% = 0.35)
- **Deal Names:** 1-500 characters
- **Email Validation:** Standard email format
- **Date Validation:** ISO 8601 format required
- **Status Transitions:** Enforced by workflow rules

Expected Data Volumes & Performance

Table	Monthly Volume	Annual Growth	Index Priority	Storage Est.
deals	500-1000	12K-15K/year	HIGH	50-100MB/year
deal_approvals	2000-5000	30K-60K/year	HIGH	20-40MB/year
deal_status_history	5000-10000	60K-120K/year	MEDIUM	15-30MB/year
approval_actions	3000-8000	40K-100K/year	MEDIUM	10-25MB/year
deal_tiers	1000-3000	15K-40K/year	HIGH	20-50MB/year
users	200-500	Steady state	LOW	1-5MB total
advertisers	100-200	500-1000/year	LOW	2-10MB/year
agencies	50-100	200-500/year	LOW	1-5MB/year

Required Indexes for Performance

High Priority Indexes (Week 1):

```
-- Deal queries (most frequent)
CREATE INDEX idx_deals_status ON deals(status);
CREATE INDEX idx_deals_created_at ON deals(created_at);
CREATE INDEX idx_deals_reference ON deals(reference_number);

-- Approval workflow queries
CREATE INDEX idx_approvals_deal_id ON deal_approvals(deal_id);
CREATE INDEX idx_approvals_assigned_to ON deal_approvals(assigned_to, status);
CREATE INDEX idx_approvals_department ON deal_approvals(department, status);

-- Audit trail queries
CREATE INDEX idx_history_deal_id ON deal_status_history(deal_id, changed_at);
```

Medium Priority Indexes (Week 3-4):

```
-- User and lookup queries
CREATE INDEX idx_users_email ON users(email);
CREATE INDEX idx_users_role_dept ON users(role, department);
CREATE INDEX idx_tiers_deal_id ON deal_tiers(deal_id, tier_number);
CREATE INDEX idx_actions_approval ON approval_actions(approval_id, created_at);
```

Technical Implementation Requirements

From system configuration (businessConstants.ts): - **Default Margin:** 35% stored as 0.35 (line 13) - **Max Tiers per Deal:** 5 (line 8) - enforced by validation - **Contract Term Default:** 12 months (line 14) - **Approval Thresholds:** $50K/$100K/$500K/$1M (lines 24-29) - **Database Type:** SQL Server compatible (T-SQL syntax) - **Authentication:** Azure AD token-based - **Performance Target:** <500ms query response - **Connection Pooling:** 10-20 connections recommended - **Transaction Isolation:** READ_COMMITTED for consistency

Migration Considerations

From In-Memory to Databricks: 1. **Data Types:** Drizzle ORM → T-SQL mapping required 2. **Enum Handling:** Convert TypeScript enums

to CHECK constraints 3. **JSON Fields:** `incentive_types` array field needs JSON support 4. **Timestamp Handling:** UTC consistency across all timestamp fields 5. **Reference Numbers:** Auto-generation sequence setup 6. **Cascade Deletes:** Ensure proper referential integrity

Schema Evolution Strategy: - Version 1.0: Core tables (deals, users, approvals) - Version 1.1: Add indexes and performance optimizations - Version 1.2: Add audit triggers and compliance features - Version 2.0: Add advanced analytics and reporting tables

---

📈 Development Timeline (6-8 Weeks)

```
Project Timeline - Deal Desk Database Implementation

Week 1-2: Infrastructure Setup
├── Anthony: Databricks environment provisioning
├── Ethan: Service account configuration
├── Ethan: Create detailed DDL scripts from schema.ts
└── 🎯 Milestone: Database connection + core tables created

Week 3-4: Implementation
├── Ethan: Deploy all 10 tables with relationships
├── Ethan: Create high-priority indexes
├── Ethan: Application integration testing
├── Ethan: Data migration from in-memory storage
└── 🎯 Milestone: App running on persistent storage with full schema

Week 5-6: Business Validation
├── Van: Trading workflow testing
├── Ethan: Performance optimization
└── 🎯 Milestone: User acceptance complete

Week 7-8: Production Deployment
├── Anthony: Production environment setup
├── Van: Trading team rollout
└── 🎯 Milestone: Live production system

Status: ⏳ Currently blocked at Week 1 - awaiting database setup
```

🤝 Infrastructure Decision Points

Anthony - Strategic Architecture Decisions Needed:

Option A: Shared Environment

```
Pros: ✅ Faster setup, shared resources
Cons: ❌ Potential conflicts, limited isolation
Cost: $ Lower
Timeline: 1-2 weeks setup
```

Option B: Dedicated Environment

```
Pros: ✅ Full control, performance isolation
Cons: ❌ More setup time, dedicated resources
Cost: $$$ Higher
Timeline: 2-3 weeks setup
```

Option C: Staged Approach

```
Pros: ✅ Test in shared, production dedicated
Cons: ❌ Two-phase migration
Cost: $$ Moderate
Timeline: 2-4 weeks total
```

Recommendation: Option C (Staged) - Start with shared test environment, migrate to dedicated production.

💡 Success Framework

Success Metrics & KPIs

Category	Metric	Target	Current Status
Data Integrity	Zero data loss events	✅ 100% persistence	❌ Blocked - in-memory only
Performance	Query response time	✅ <500ms average	⚠️ TBD - needs testing
Scalability	Concurrent users	✅ 200+ supported	❌ Limited to single session
Reliability	System uptime	✅ 99.9% availability	❌ Restart = data loss
Compliance	Complete audit trail	✅ Full history tracking	❌ No persistence
User Adoption	Trading team usage	50+ users within 30 days	⏳ Pending database setup

Business Success Metrics

Business Goal	Success Criteria	Measurement
Operational Efficiency	40% reduction in deal approval cycle time	Average time from submission to approval
Pipeline Visibility	Real-time tracking across all approval tiers	Dashboard usage and deal status accuracy
Compliance Readiness	Complete audit trail for regulatory requirements	100% of status changes logged with user attribution
Team Adoption	50+ active users within 30 days of launch	Daily/weekly active user metrics

---

🚀 Immediate Action Items

This Week (Critical Path - Ethan & Anthony)

Anthony's Decisions Needed: - [ ] Choose environment approach (A, B, or C above) - [ ] Approve test database provisioning
- [ ] Assign technical contact for setup coordination

Ethan's Deliverables: - [ ] Provide detailed schema DDL scripts - [ ] Document connection requirements - [ ] Create migration testing plan

Next Week

Joint Activities: - [ ] Database environment validation - [ ] Initial connection testing - [ ] Schema deployment verification

---

🔧 Technical Specifications Detail

Connection Requirements

```
Database Configuration Needed:
├── Server: [TBD - Anthony to provide]
├── Database: deal_desk_test (initial)
├── Auth: Azure AD service account
├── Permissions: CREATE, SELECT, INSERT, UPDATE, DELETE
├── SSL: Required (TLS 1.2+)
└── Pooling: Connection pool size: 10-20
```

Service Account Setup

- **Account Name:** `svc-dealdesk-prod`
- **Permissions:** Read/write to designated schema only
- **Rotation:** Standard MiQ security policy compliance
- **Monitoring:** Query performance and access logging

📞 Coordination & Communication

Immediate Team Sync Needed: - **Anthony & Ethan:** Technical architecture review (30 min) - **Van & Ethan:** Business validation planning (30 min)
- **All Three:** Weekly status check-ins during 6-8 week timeline

Questions for Discussion: 1. Which environment approach fits Innovation & Growth Programs strategy? 2. Any existing Databricks infrastructure we can leverage? 3. Standard process for promoting test → production in our department?

📋 Risk Mitigation

Risk	Impact	Mitigation
Data Loss	🔴 HIGH	Immediate database setup
Performance	🟡 MEDIUM	Load testing in week 4-5
Integration	🟡 MEDIUM	Parallel development streams
Trading Team Adoption	🟠 MEDIUM	Van's change management plan

Next Steps: Anthony, please review the environment options above and let's schedule a brief technical planning session. Van's trading team is ready to validate the solution as soon as we have persistent storage.

Internal Contacts: - **Ethan Sam** (Technical): ethan.sam@miqdigital.com - **Van Ngo** (Business): van.ngo@miqdigital.com