

Database Issues and Recommended Fixes

PSScript Platform - January 15, 2026 Audit

Summary

Priority	Count	Status
Critical	2	Needs Immediate Fix
High	5	Fix This Sprint
Medium	8	Backlog
Low	4	Nice to Have

Critical Issues

CRIT-001: Missing Exports in Database Connection Module

File: `src/backend/src/models/index.ts`

Issue: The models index file imports `dbConnectionInfo` and `connectionEvents` from the connection module, but these exports don't exist in `src/backend/src/database/connection.ts`.

```
// Current (BROKEN):  
import { dbConnectionInfo, connectionEvents } from '../database/co  
  
// connection.ts does NOT export these
```

Impact: - Application may crash on import - TypeScript errors silently ignored due to `@ts-nocheck`

Fix:

```
// Option 1: Add exports to connection.ts  
export const dbConnectionInfo = {  
  host: process.env.DB_HOST,  
  port: process.env.DB_PORT,  
  database: process.env.DB_NAME,  
  poolSize: POOL_MAX  
};  
  
export const connectionEvents = new EventEmitter();  
  
// Option 2: Remove unused imports from index.ts  
// Remove the import line entirely if not needed
```

Priority: CRITICAL **Effort:** 30 minutes **Risk if unfixed:** Application instability

CRIT-002: Schema Mismatch Between SQL and Models

Issue: The base `schema.sql` doesn't include several columns that migrations add, creating potential sync issues on fresh installs.

Missing from schema.sql: - `scripts.file_hash` (added by migration) - `users.last_login_at` (added by migration) - `users.login_attempts` (added by migration) - `users.locked_until` (added by migration) - Extended `script_analysis` JSONB columns

Impact: - Fresh database setup may have incomplete schema - Migration order dependencies - Docker initialization inconsistency

Fix: Update `src/db/schema.sql` to include all current columns:

```
-- Add to users table:
last_login_at TIMESTAMP WITH TIME ZONE,
login_attempts INTEGER DEFAULT 0,
locked_until TIMESTAMP WITH TIME ZONE,

-- Add to scripts table:
file_hash VARCHAR(255),

-- Update script_analysis with all JSONB fields
```

Priority: CRITICAL **Effort:** 2 hours **Risk if unfixed:** Data inconsistency on new deployments

High Priority Issues

HIGH-001: Missing CASCADE on User-Script Relationship

File: `src/db/schema.sql`

Issue: The `scripts.user_id` foreign key doesn't specify cascade behavior.

```
-- Current:
user_id INTEGER REFERENCES users(id),

-- Should be:
user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,
```

Impact: Orphaned scripts when users are deleted

Fix: Add migration to alter foreign key constraint

Priority: HIGH **Effort:** 1 hour

HIGH-002: Missing Index on file_hash Column

File: Index configuration

Issue: The `file_hash` column is used for deduplication queries but may not have an index in the migration.

Current Query Pattern:

```
SELECT * FROM scripts WHERE file_hash = 'abc123...'
```

Fix:

```
CREATE INDEX IF NOT EXISTS idx_scripts_file_hash ON scripts(file_
```

Priority: HIGH **Effort:** 15 minutes

HIGH-003: Inadequate Connection Pool Size for Production

File: `src/backend/src/database/connection.ts`

Issue: Pool max of 10 connections may be insufficient for production traffic.

Current:

```
const POOL_MAX = 10;
```

Recommendation: - Development: 5-10 - Staging: 15-20 - Production: 25-50
(based on load testing)

Fix:

```
const POOL_MAX = parseInt(process.env.DB_POOL_MAX || '10');
```

Priority: HIGH **Effort:** 30 minutes

HIGH-004: No Index on chat_history.embedding Column

Issue: Vector similarity searches on chat history will be slow without an index.

Fix:

```
CREATE INDEX IF NOT EXISTS chat_history_embedding_idx  
ON chat_history USING ivfflat (embedding vector_cosine_ops)  
WITH (lists = 100);
```

Priority: HIGH **Effort:** 30 minutes

HIGH-005: Raw SQL Queries in Analytics Controller

File: Analytics routes

Issue: Mixed use of Sequelize ORM and raw SQL queries reduces consistency and may have SQL injection vectors.

Example:

```
// Raw SQL mixed with ORM
await sequelize.query(`SELECT ... FROM script_analysis ...`);
```

Fix: Convert to Sequelize query methods or ensure parameterization

Priority: HIGH **Effort:** 4 hours

Medium Priority Issues

MED-001: Missing updatedAt Trigger for Automatic Timestamps

Issue: PostgreSQL doesn't automatically update `updated_at` columns. Sequelize handles this, but raw SQL updates won't.

Fix:

```
CREATE OR REPLACE FUNCTION update_updated_at_column()
RETURNS TRIGGER AS $$
BEGIN
    NEW.updated_at = CURRENT_TIMESTAMP;
    RETURN NEW;
END;
$$ language 'plpgsql';

CREATE TRIGGER update_scripts_updated_at
    BEFORE UPDATE ON scripts
    FOR EACH ROW
    EXECUTE FUNCTION update_updated_at_column();
```

Priority: MEDIUM **Effort:** 1 hour

MED-002: No Soft Delete Implementation

Issue: All deletes are hard deletes, making recovery impossible.

Recommendation: Add `deleted_at` column for soft deletes on critical tables (scripts, users).

```
ALTER TABLE scripts ADD COLUMN deleted_at TIMESTAMP WITH TIME ZONE;
CREATE INDEX idx_scripts_deleted_at ON scripts(deleted_at);
```

Priority: MEDIUM **Effort:** 4 hours

MED-003: Script Content Not Compressed

Issue: Large PowerShell scripts stored as plain TEXT consume significant storage.

Recommendation: Consider TOAST compression or application-level compression for scripts > 10KB.

Priority: MEDIUM **Effort:** 8 hours

MED-004: No Database Backup Automation

Issue: No automated backup strategy documented or implemented.

Fix: Add pg_dump scheduled backup:

```
# Add to cron (daily at 2 AM)
0 2 * * * pg_dump -h localhost -U postgres psscript | gzip > /back
```

Priority: MEDIUM **Effort:** 2 hours

MED-005: Missing Unique Constraint on ScriptAnalysis

Issue: One-to-one relationship between Script and ScriptAnalysis not enforced at DB level.

Fix:

```
ALTER TABLE script_analysis ADD CONSTRAINT script_analysis_script
```

Priority: MEDIUM **Effort:** 30 minutes

MED-006: execution_logs Growing Unbounded

Issue: No retention policy for execution logs. Table will grow indefinitely.

Fix: Add partition by date or scheduled cleanup:

```
-- Delete logs older than 90 days
DELETE FROM execution_logs WHERE created_at < NOW() - INTERVAL '90 days'
```

Priority: MEDIUM **Effort:** 2 hours

MED-007: Redis Cache Key Collision Risk

Issue: Cache key patterns may collide across different filter combinations.

Current:

```
const cacheKey = `scripts:${page}:${limit}:${categoryId}`;
```

Fix: Use deterministic hashing for complex filter combinations:

```
const filterHash = crypto.createHash('md5').update(JSON.stringify(filters)).digest('hex');
const cacheKey = `scripts:${filterHash}`;
```

Priority: MEDIUM **Effort:** 2 hours

MED-008: No Connection Monitoring/Alerting

Issue: No metrics collection for database connection health.

Recommendation: Add Prometheus metrics: - Connection pool utilization - Query execution time histogram - Error rate by type

Priority: MEDIUM **Effort:** 8 hours

Low Priority Issues

LOW-001: Inconsistent Timestamp Column Names

Issue: Mixed naming conventions: `created_at` vs `createdAt`, `updated_at` vs `updatedAt`.

Tables with underscore: PostgreSQL schema **Tables with camelCase:** Sequelize defaults

Fix: Standardize on one convention (underscore recommended for PostgreSQL)

Priority: LOW **Effort:** 4 hours

LOW-002: No Database Comments on All Tables

Issue: Only `script_analysis` has column comments. Other tables lack documentation.

Fix:

```
COMMENT ON TABLE scripts IS 'PowerShell scripts uploaded by users'
COMMENT ON COLUMN scripts.file_hash IS 'MD5 hash for deduplication'
```

Priority: LOW **Effort:** 2 hours

LOW-003: @ts-nocheck in Models Index

Issue: TypeScript checking disabled, hiding potential type errors.

File: `src/backend/src/models/index.ts`

```
// @ts-nocheck – Required for circular model references
```

Fix: Properly type associations to remove `@ts-nocheck`

Priority: LOW **Effort:** 8 hours

LOW-004: Missing Documentation Model Validation

Issue: URL length (2048 chars) may be insufficient for some Microsoft Learn URLs with query strings.

Fix: Consider TEXT type or validate URL length on input

Priority: LOW **Effort:** 1 hour

Performance Optimization Recommendations

PERF-001: Add Partial Indexes for Common Queries

```
-- Only index public scripts (most common query)
CREATE INDEX idx_scripts_public ON scripts(created_at) WHERE is_public

-- Only index active users (not locked)
CREATE INDEX idx_users_active ON users(last_login_at) WHERE locked = false
```

PERF-002: Consider Read Replicas

For production with > 1000 concurrent users, implement read replicas for: - Script listings - Search queries - Analytics

PERF-003: Implement Query Result Caching

Current caching covers basic lists. Extend to: - Search results (with short TTL) - Analytics aggregations (longer TTL) - User profiles

PERF-004: Optimize Vector Search

Current IVFFlat index with 100 lists. For > 100,000 scripts: - Increase lists to 300-500 - Consider HNSW index for better recall - Implement approximate search with result reranking

Migration Checklist for Fixes

Sprint 1 (Immediate)

- ☐ CRIT-001: Fix connection module exports
- ☐ CRIT-002: Update schema.sql with all columns
- ☐ HIGH-002: Add file_hash index

Sprint 2 (This Week)

- ☐ HIGH-001: Add CASCADE to foreign keys
- ☐ HIGH-003: Make pool size configurable
- ☐ HIGH-004: Add chat_history embedding index

Sprint 3 (This Month)

- ☐ HIGH-005: Refactor raw SQL queries
- ☐ MED-001: Add updated_at triggers
- ☐ MED-005: Add unique constraint on script_analysis

Backlog

- ☐ MED-002: Implement soft deletes
 - ☐ MED-006: Add execution_logs retention
 - ☐ All LOW priority items
-

Testing Requirements

After implementing fixes, verify:

1. **Schema Consistency** `bash # Compare schema.sql with actual database pg_dump -h localhost -U postgres --schema-only psscript > actual_schema.sql diff schema.sql actual_schema.sql`
2. **Index Verification** `sql SELECT indexname, indexdef FROM pg_indexes WHERE tablename IN ('scripts', 'users', 'script_analysis');`
3. **Foreign Key Integrity** `sql SELECT conname, conrelid::regclass, confrelid::regclass FROM pg_constraint WHERE contype = 'f';`
4. **Performance Baseline**
5. Run query performance tests before/after index changes
6. Document response times for common queries

Audit completed: January 15, 2026 Next audit scheduled: April 15, 2026 Auditor: Claude Code (Opus 4.5)