

# **PSScript Database Infrastructure**

## **Update - January 16, 2026**

---

# Change Record

Document ID	DCR-2026-016
Version	1.0
Date	January 16, 2026
Author	Database Infrastructure Team
Review Status	Completed
Effective Date	January 16, 2026

# Executive Summary

---

**Status:** Infrastructure improvements completed successfully **Test Results:** 81/81 tests passing (100%) **Changes:** 4 indexes created, 3 Sequelize models added, 1 frontend monitoring component

This document records database infrastructure improvements made on January 16, 2026, building upon the comprehensive review completed on January 15, 2026 (see DATABASE\_REVIEW\_2026.md).

---

# Changes Implemented

---

## 1. Foreign Key Index Optimization

**Issue Identified:** 4 foreign keys lacked supporting indexes, causing potential query performance degradation during JOIN operations.

**Resolution:** Created indexes on all unindexed foreign keys:

```
CREATE INDEX idx_chat_history_user ON chat_history(user_id);
CREATE INDEX idx_comments_user ON comments(user_id);
CREATE INDEX idx_comments_script ON comments(script_id);
CREATE INDEX idx_script_versions_user ON script_versions(user_id);
```

**Impact:** - Improved JOIN performance for user-related queries - Better query planner estimates for foreign key lookups - Expected 10-50% improvement on queries involving these tables

## 2. New Sequelize Models

**Issue Identified:** 3 database tables existed without corresponding Sequelize models, creating inconsistent data access patterns.

**Resolution:** Created type-safe Sequelize models with proper associations:

### Comment Model ( `src/backend/src/models/Comment.ts` )

- Maps to `comments` table
- Associations: belongsTo User, belongsTo Script
- Indexes defined at model level for consistency

### UserFavorite Model ( `src/backend/src/models/UserFavorite.ts` )

- Maps to `user_favorites` junction table
- Composite primary key (user\_id, script\_id)
- Associations: belongsTo User, belongsTo Script
- No updatedAt timestamp (by design)

## ScriptDependency Model ( `src/backend/src/models/ScriptDependency.ts` )

- Maps to `script_dependencies` table
- Composite primary key (parent\_script\_id, child\_script\_id)
- Associations: belongsTo Script (both parent and child)
- Tracks script-to-script dependency relationships

**Model Registration:** All models registered in

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

## 3. Table Maintenance

**Issue Identified:** Table bloat detected during infrastructure analysis: -

`script_versions` : 1400% dead row ratio - `script_analysis` : 1600% dead row ratio

**Resolution:** Executed VACUUM ANALYZE on all tables:

```
VACUUM ANALYZE;
```

**Impact:** - Reclaimed disk space from dead tuples - Updated query planner statistics - Improved query performance estimates

## 4. Frontend Monitoring Component

**Component:** `DatabaseAdminPanel` ( `src/frontend/src/components/admin/` )

**Features:** - Real-time connection pool monitoring - Table health visualization with status indicators - Index usage statistics display - Manual VACUUM trigger capability - Industrial control room aesthetic design

**Files Created:** - `DatabaseAdminPanel.tsx` - React component - `DatabaseAdminPanel.css` - Styling - `index.ts` - Barrel export

**Dependencies Added:** - `framer-motion` - Animation library for component transitions

---

# Updated Schema Summary

## Tables: 15 Total

Table	Sequelize Model	Status
users	User	OK
scripts	Script	OK
categories	Category	OK
tags	Tag	OK
script_tags	ScriptTag	OK
script_versions	ScriptVersion	OK
script_analysis	ScriptAnalysis	OK
script_embeddings	ScriptEmbedding	OK
execution_logs	ExecutionLog	OK
chat_history	ChatHistory	OK
documentation	Documentation	OK
comments	<b>Comment</b>	<b>NEW</b>
user_favorites	<b>UserFavorite</b>	<b>NEW</b>
script_dependencies	<b>ScriptDependency</b>	<b>NEW</b>
agent_state	(pending)	REVIEW

## Indexes: 44+ Total

**New Indexes Created:** | Index Name | Table | Column(s) | |-----|-----|-----  
---| | idx\_chat\_history\_user | chat\_history | user\_id | | idx\_comments\_user |  
comments | user\_id | | idx\_comments\_script | comments | script\_id | |  
idx\_script\_versions\_user | script\_versions | user\_id |



# Connection Pool Status

Metric	Value	Status
Active Connections	3	OK
Pool Maximum	100	OK
Utilization	3%	OK
Idle Timeout	10,000ms	OK



# Verification Results

---

## Test Suite

```
Test Suites: 6 passed, 6 total
Tests:      81 passed, 81 total
Time:       5.126s
```

## TypeScript Compilation

```
npx tsc --noEmit
# No errors
```

## Model Association Verification

All new models successfully: - Initialize with Sequelize instance - Define correct associations - Map to existing database schema

---

# Sprint Progress Update

---

From DATABASE\_REVIEW\_2026.md recommendations:

## Sprint 2 Items (Addressed Today)

- ☒ Create Sequelize models for missing tables (3 of 3 completed)
- ☐ Fix N+1 queries with eager loading (pending)
- ☐ Increase pool min to 2 (pending)

## Additional Improvements

- ☒ Index optimization for foreign keys
  - ☒ Table maintenance (VACUUM ANALYZE)
  - ☒ Monitoring dashboard component
-

# Risk Assessment

---

Change	Risk Level	Mitigation
New indexes	Low	Non-blocking CREATE INDEX
New models	Low	No schema changes, models match existing tables
VACUUM ANALYZE	Low	Standard maintenance operation
Frontend component	None	Isolated admin feature

---

# Rollback Procedures

---

If issues arise:

## Indexes

```
DROP INDEX IF EXISTS idx_chat_history_user;  
DROP INDEX IF EXISTS idx_comments_user;  
DROP INDEX IF EXISTS idx_comments_script;  
DROP INDEX IF EXISTS idx_script_versions_user;
```

## Models

Remove from `models/index.ts`: - Comment import and initialization -  
UserFavorite import and initialization - ScriptDependency import and initialization

## Frontend Component

Remove directory: `src/frontend/src/components/admin/`

---

## Next Steps

---

1. **Commit Changes:** Stage and commit all new files
  2. **Continue Sprint 2:** Address N+1 query patterns
  3. **Pool Optimization:** Increase min pool size to 2
  4. **Agent Models:** Create Sequelize models for agent\_state, conversation\_history, tool\_execution\_results
-

## Appendix: New Model Schemas

---

### Comment Model

```
interface Comment {  
  id: number;           // Primary key  
  scriptId: number;     // FK -> scripts.id (CASCADE DELETE)  
  userId: number;       // FK -> users.id  
  content: string;      // Comment text  
  createdAt: Date;  
  updatedAt: Date;  
}
```

### UserFavorite Model

```
interface UserFavorite {  
  userId: number;       // PK, FK -> users.id  
  scriptId: number;     // PK, FK -> scripts.id (CASCADE DELETE)  
  createdAt: Date;  
}
```

### ScriptDependency Model

```
interface ScriptDependency {  
  parentScriptId: number; // PK, FK -> scripts.id (CASCADE DELETE)  
  childScriptId: number;  // PK, FK -> scripts.id (CASCADE DELETE)  
  createdAt: Date;  
}
```

---

*Document Control: DCR-2026-016 Review Cycle: As needed upon infrastructure changes Related Documents: DATABASE\_REVIEW\_2026.md*