

Migration Fix Summary - December 21, 2025

Issue

Error: P3009 - Failed migration blocking deployment

Migration: 20251221011617_add_document_classification_validation

Status: FAILED during deployment

Impact: Blocking all new deployments

Root Cause Analysis

Primary Issue

The migration 20251221011617_add_document_classification_validation failed during deployment because:

1. **Previous Failed Migrations:** The database has failed migrations in its history:
 - draft_add_document_processing - Failed trying to create DocumentType enum that already exists
 - 20251218162945_update_homes_to_active - Previously failed and rolled back
 - 20251221011617_add_document_classification_validation - Latest failure
2. **Prisma Behavior:** Prisma refuses to apply new migrations when there are failed migrations in the database history (Error P3009)
3. **Migration Chain Blocking:** Even though the migration SQL uses safe patterns (DO blocks, exception handling), it cannot execute until previous failed migrations are resolved

Technical Details

Error from Render Logs (Dec 20, 2025):

```
Error: P3018
A migration failed to apply. New migrations cannot be applied before the error is recovered from.

Migration name: draft_add_document_processing
Database error code: 42710
Database error: ERROR: type "DocumentType" already exists
```

Migration File Analysis

The 20251221011617_add_document_classification_validation migration includes:

- Enum type updates (DocumentType, ValidationStatus, ReviewStatus)
- New columns for document classification
- Indexes for performance
- Foreign key relationships

The migration uses safe patterns:

- DO \$\$ BEGIN ... EXCEPTION blocks for idempotency

- `IF NOT EXISTS` checks
- Proper enum value mapping

However, it cannot execute because Prisma blocks it due to previous failures.

Solution Implemented

Fix Details

Updated the `migrate:deploy` script in `package.json` to roll back ALL failed migrations before deploying:

Before:

```
"migrate:deploy": "npx prisma migrate resolve --rolled-back
draft_add_document_processing || true && npx prisma migrate resolve --rolled-back
20251218162945_update_homes_to_active || true && npx prisma migrate deploy"
```

After:

```
"migrate:deploy": "npx prisma migrate resolve --rolled-back
draft_add_document_processing || true && npx prisma migrate resolve --rolled-back
20251218162945_update_homes_to_active || true && npx prisma migrate resolve --rolled-
back 20251221011617_add_document_classification_validation || true && npx prisma
migrate deploy"
```

Why This Works

- Rollback First:** Marks failed migrations as “rolled back” in the `_prisma_migrations` table
 - Error Suppression:** `|| true` ensures the script continues even if migrations are already rolled back
 - Clean Slate:** After rollbacks, `prisma migrate deploy` can apply migrations from a clean state
 - Idempotent:** Can be run multiple times safely
-

Changes Made

Files Modified

- package.json**
 - Updated `migrate:deploy` script
 - Added rollback for `20251221011617_add_document_classification_validation`

Git Commit

- **Commit:** 6385e60
 - **Branch:** main
 - **Pushed:** Yes 
-

Deployment

Automatic Deployment

- ✅ Fix committed and pushed to GitHub
- 🚀 Render auto-deployment triggered
- ⏱ ETA: 5-10 minutes

What Happens on Next Deploy

- Pre-deploy Phase:** `npm run migrate:deploy` executes
 - Rollback Migrations:** All three failed migrations marked as rolled back
 - Apply Migrations:** Prisma applies any pending migrations cleanly
 - Start Application:** Next.js app starts with updated schema
-

Verification Steps

1. Check Render Deployment Logs

Look for these successful messages:

```
Migration draft_add_document_processing marked as rolled back.
Migration 20251218162945_update_homes_to_active marked as rolled back.
Migration 20251221011617_add_document_classification_validation marked as rolled back.

Applying migration `20251221011617_add_document_classification_validation`
✓ Migration applied successfully
```

2. Verify Database Schema

After deployment, check that Document table has new columns:

```
SELECT column_name, data_type
FROM information_schema.columns
WHERE table_name = 'Document'
AND column_name IN (
    'documentType',
    'classificationConfidence',
    'classificationReasoning',
    'autoClassified',
    'validationStatus',
    'reviewStatus',
    'reviewedById',
    'reviewedAt'
)
ORDER BY column_name;
```

3. Test Application Features

- ✅ Document upload works
 - ✅ Document classification functions
 - ✅ Phase 3 features operational
-

Technical Notes

Migration Strategy

This migration uses a **safe enum update strategy**:

1. Create new enum type `DocumentType_new`
2. Alter table to use new type with value mapping
3. Drop old enum and rename new one

This avoids issues with:

- Existing enum values in use
- Data migration during enum changes
- Transaction conflicts

Future Prevention

To prevent similar issues:

1. **Test migrations locally** before deploying
 2. **Use idempotent SQL** (DO blocks, IF NOT EXISTS)
 3. **Monitor Prisma migration status** in production
 4. **Keep rollback scripts updated** as new migrations are added
-

Related Migrations

Phase 1A Migrations (Dec 20, 2025)

- 20251220025013_phase1aEnums : Added enum types and values
- 20251220025039_phase1a_columns_and_tables : Added columns and tables

Phase 3 Part 1 Migration (Dec 21, 2025)

- 20251221011617_add_document_classification_validation : Document classification and validation features
-

Status

- **Issue Identified**: Failed migration blocking deployment
 - **Root Cause Found**: Previous failed migrations in database
 - **Solution Implemented**: Updated rollback script
 - **Fix Committed**: Commit 6385e60
 - **Fix Pushed**: To `main` branch
 - **Deployment**: Auto-triggered on Render
 - **Verification**: Pending deployment completion
-

Next Steps

1. **Monitor Render Deployment**: Watch logs for successful migration
2. **Verify Schema**: Confirm new columns exist in Document table

3. **Test Features:** Validate Phase 3 document classification works
 4. **Update Documentation:** Document any additional issues discovered
-

Last Updated: December 21, 2025

Status:  FIXED - Awaiting Deployment Verification