DOCUPAL
Docupal Demo, LLC

Table of Contents

Introduction

Docupal Demo, LLC presents this Yii Migration Proposal to ACME-1 for database schema updates. The goal is to enhance the user management module. This will be achieved by upgrading the database schema.

Purpose of Migration

The current database lacks efficient support for storing user roles and permissions. This migration addresses this limitation. The new schema will allow ACME-1 to manage user access more effectively. This ensures better security and control.

Project Phase

This migration is part of Phase 2: Database Modernization of the overall project. It focuses specifically on the database aspects of the user management module upgrade.

Overview

This proposal details the plan to migrate the database schema. It includes objectives, impacted tables, and schema changes. It also covers data transformation, dependencies, and migration tools. Furthermore, the proposal describes the deployment process, rollback plan, and risk assessment. Finally, it outlines testing procedures, the communication strategy, and required approvals.

Current Database Schema Overview

The current database schema focuses on user accounts and their associated data. This migration directly impacts the users, roles, and permissions tables.

Users Table

The users table stores user account information. It contains a role_id field, which will be modified to support the new roles and permissions system. A foreign key constraint exists on users.account_id, linking each user to an account. There is an index on the users.email field for efficient lookups. This table has a dependency on the accounts table.

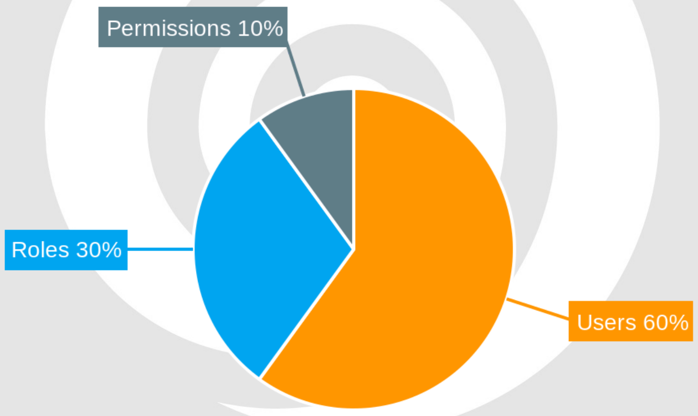
Roles Table

The roles table currently stores role names in the roles.name field. The migration will modify this table to accommodate a more granular permission structure.

Permissions Table

The permissions table stores permission names in the permissions.name field. It will be updated to align with the new roles and permissions system.

It may be helpful to visualize the schema using a diagram. Consider using an external tool to generate a schema diagram illustrating the relationships between the users, roles, and permissions tables, and their connection to the accounts table.



Proposed Migration Details

This section details the proposed changes to the database schema. The migration aims to enhance user role and permission management within the ACME-1 application. We will add new tables and modify an existing one. We will also handle the transformation and movement of existing data.

Schema Modifications

The migration will introduce the following new tables:

- roles: This table will store the different user roles (e.g., administrator, editor, viewer).
- permissions: This table will define specific permissions within the application (e.g., create posts, edit users, view reports). A many-to-many relationship between roles and permissions will be established.

The existing users table will be modified to include a role_id column. This column will be a foreign key referencing the roles table, linking each user to a specific role.

Data Transformation and Movement

Existing user type data will be transformed to fit the new role-based system. This involves mapping current user types to the new roles defined in the roles table. For example, a user with type "admin" might be mapped to the "administrator" role. This mapping will be carefully defined and documented.

The roles and permissions tables will be populated with default values during the migration. These default values will represent the standard roles and permissions required for the application to function correctly. This ensures that the system is operational immediately after the migration.

Dependencies and Potential Conflicts

This migration has some dependencies. Application code that relies on the old user type system will need to be updated to use the new role-based system. This will involve code changes to check user roles instead of user types.

Potential conflicts may arise if existing user type values do not map directly to the new roles. This requires careful analysis and potentially the creation of new roles to accommodate these values. We will work closely with ACME-1 to resolve these mapping issues.

Migration Change Impact

The following chart illustrates the impact of the migration across different components over time.

Migration Implementation Plan

The database migration will be executed in a phased approach across our development, staging, and production environments. We will use the Yii migration tool to apply the schema changes. Custom PHP scripts will handle data transformations.

Migration Steps

1. **Development Environment:**
   - Apply migrations to the local development database using the yii migrate command.
   - Run data transformation scripts.
   - Perform thorough testing to validate the changes.
2. **Staging Environment:**
   - Update the staging environment database.
   - Apply migrations using the yii migrate command.
   - Execute data transformation scripts.
   - Conduct user acceptance testing with ACME-1 representatives.
3. **Production Environment:**
   - Take a full database backup.
   - Apply migrations to the production database using the yii migrate command.
   - Run data transformation scripts.
   - Monitor the application closely for any issues.

Rollback Strategy

If the migration fails, we will revert the changes. This involves the following:

1. Immediately stop the migration process.
2. Use the yii migrate down command to revert the applied migrations.
3. Restore the database to the pre-migration state using the backup created before starting.

4. Investigate the cause of the failure.
5. Implement necessary corrections.
6. Repeat the migration process in a controlled manner.

Tools and Scripts

- **Yii Migration Tool:** For applying and reverting schema changes.
- **Custom PHP Scripts:** For transforming existing data to fit the new schema.
- **Database Backup Tools:** For creating and restoring database backups.

Risk Assessment and Mitigation

This section identifies potential risks associated with the Yii migration and outlines mitigation strategies to minimize their impact.

Potential Risks

- **Data Loss:** There is a risk of data loss during the transformation and migration process. This could result from errors in the migration scripts or unexpected issues during data transfer.
- **Migration Script Errors:** Errors in the migration scripts could lead to data corruption or incomplete schema changes.
- **Application Incompatibility:** The application might become incompatible with the database schema after the migration. This could result in application errors or downtime.

Mitigation Strategies

To ensure data integrity and a smooth migration, we will implement the following strategies:

- **Data Validation Scripts:** We will create and execute data validation scripts before and after the migration. These scripts will verify data integrity and identify any discrepancies.
- **Database Backups:** We will perform full database backups before initiating the migration. This will provide a fallback option in case of critical failures.
- **Transaction Management:** We will use transaction management during the migration process. This ensures that all schema changes and data transformations are applied atomically, minimizing the risk of data corruption.

- **Rollback Plan:** In case of failure, we can restore the database from the latest backup. We will also prepare manual data correction procedures for minor issues.
- **Testing:** We will conduct thorough testing of the application after the migration to identify and resolve any incompatibility issues.

Testing Strategy

Our testing strategy ensures a smooth and reliable Yii migration for ACME-1. The Quality Assurance Team will spearhead all testing efforts. We will conduct tests both before and after the migration.

Pre-Migration Testing

Before the migration, we will focus on the accuracy of our data transformation scripts. We will achieve this by:

- **Unit Tests:** Rigorous unit tests will validate each script. These tests will confirm that the scripts correctly transform data according to the new schema.

Post-Migration Testing

After the migration, our focus will shift to ensuring that the application functions as expected with the new database schema. Our approach includes:

- **Integration Tests:** We will perform extensive integration tests. These tests will verify that all application components work seamlessly with the migrated database. These tests will include data integrity checks and application functionality verification.
- **Schema Verification:** We will verify that the database schema matches the intended design.
- **Data Integrity Checks:** We will check that the data has been migrated correctly and that no data has been lost or corrupted.
- **Application Functionality Tests:** We will test all the key features of the application to ensure that they are working as expected.

Measurement and Tracking

We will carefully measure and track the results of all tests. Success will be determined by:

- **Schema Changes:** Validating the successful implementation of all schema modifications.
- **Data Integrity:** Confirming that all data has been migrated accurately and completely.
- **Application Functionality:** Verifying that the application functions correctly with the new database.

Deployment and Rollback Procedures

The deployment of the Yii migration will occur over a 5-day period. This timeline includes development, testing, staging, user acceptance testing (UAT), and the final production deployment.

Deployment Process

The deployment process will follow these steps:

1. **Day 1: Development.** The migration scripts will be developed and tested in a development environment.
2. **Day 2: Testing.** Comprehensive testing of the migration scripts will be performed.
3. **Day 3: Staging.** The migration will be deployed to a staging environment for final validation.
4. **Day 4: User Acceptance Testing (UAT).** Key users will perform UAT to ensure the application functions as expected with the migrated data.
5. **Day 5: Production Deployment.** The migration will be applied to the production database.

Rollback Procedures

Rollback will be initiated if critical errors are found during post-migration testing or if the application does not function correctly after the migration.

The rollback procedure involves running the 'down' migration scripts to revert the database schema to its previous state. A database backup taken before the migration will also be available as a secondary rollback option.

Steps for Rollback:

1. **Identify the Issue.** Determine the cause and severity of the problem.
2. **Trigger Rollback.** Initiate the rollback process if the issue is critical.
3. **Execute Rollback Scripts.** Run the Yii migration rollback commands to revert the schema changes.
4. **Restore Backup (if needed).** If the rollback scripts fail or are insufficient, restore the database from the pre-migration backup.
5. **Verify Rollback.** Confirm that the database has been successfully reverted to its original state.

Communication

Email notifications will be sent to stakeholders to communicate the deployment schedule and any potential impacts. The project status page will be updated regularly to provide progress updates and any relevant information.

Impact on Application and Users

This database migration will introduce changes that impact both the application and its users.

Application Impact

The application code will require updates to align with the new roles and permissions system. Developers will need to modify the codebase to leverage the updated schema, ensuring that user roles and permissions are correctly enforced throughout the application. This includes changes to authentication, authorization, and data access layers.

User Impact

Users will experience a brief downtime of approximately 15 minutes during the production migration. This is necessary to ensure data integrity during the schema update. To minimize disruption, we will schedule the migration during off-peak hours.

Prior to the migration, users will be notified via email and an in-application notification banner. This communication will inform them about the upcoming downtime and any potential impact on their access to the application.

Post-migration, users should experience no immediate changes in functionality. However, the new roles and permissions system will allow for more granular control over access to features and data, which may result in changes to individual user permissions over time. These changes will be communicated to affected users as they are implemented.

Approval and Sign-Off

This Yii Migration Proposal requires formal approval before execution. The following stakeholders are responsible for reviewing and approving this document: the Lead Developer, the Database Administrator, and the Project Manager.

Required Documentation

To facilitate the approval process, the following documentation must be provided for review:

- Migration script
- Database schema diagrams
- Test results

Approval Recording

All approvals will be recorded within the project management system. This ensures a clear audit trail and accountability for the migration process. The sign-off criteria include verification that the migration script is syntactically correct and aligns with the proposed schema changes, the database schema diagrams accurately reflect the intended state, and the test results demonstrate successful migration and data integrity.