

Contract Management System - Database Documentation

Table of Contents

1. [Overview](#)
2. [Database Schema](#)
3. [Table Descriptions](#)
4. [Business Logic](#)
5. [Views](#)
6. [Stored Procedures](#)
7. [Triggers](#)
8. [Usage Examples](#)
9. [Security Considerations](#)
10. [Performance Optimization](#)

Overview

The Contract Management System (CMS) is a comprehensive database solution designed to manage the complete lifecycle of contracts within an organization. The system supports contract creation, approval workflows, document management, renewals, amendments, notifications, and comprehensive audit trails.

Key Features

- **User Role Management:** Admin, Creator, Approver, and Viewer roles
- **Contract Lifecycle:** Draft → Pending Approval → Approved/Rejected → Active → Expired/Terminated
- **Document Management:** Multiple document types with versioning
- **Audit Trail:** Complete history of all contract changes
- **Automated Notifications:** Expiry warnings and renewal reminders
- **Contract Renewals:** Automatic and manual renewal processes
- **Amendments:** Track contract modifications with approval workflows
- **Soft Delete:** Data preservation with logical deletion

Database Schema

Entity Relationship Overview



Table Descriptions

Users Table

Purpose: Manages system users and their roles

Column	Type	Description
user_id	INT (PK)	Unique user identifier
username	VARCHAR(50)	Unique username for login
email	VARCHAR(100)	User's email address
password_hash	VARCHAR(255)	Hashed password
first_name, last_name	VARCHAR(50)	User's full name
role	ENUM	Admin, Creator, Approver, Viewer
department	VARCHAR(100)	User's department
is_active	BOOLEAN	Account status
created_at, updated_at	TIMESTAMP	Audit timestamps

Role Permissions:

- **Admin:** Full system access, user management
- **Creator:** Create and edit contracts
- **Approver:** Approve/reject contracts
- **Viewer:** Read-only access

Contracts Table

Purpose: Core contract information and status tracking

Column	Type	Description
contract_id	INT (PK)	Unique contract identifier
contract_number	VARCHAR(50)	Business contract number
title	VARCHAR(200)	Contract title
description	TEXT	Detailed description
contract_type	ENUM	Service, Purchase, Employment, Lease, NDA, Other
party_a, party_b	VARCHAR(200)	Contract parties
start_date, end_date	DATE	Contract duration
contract_value	DECIMAL(15,2)	Monetary value
currency	VARCHAR(3)	Currency code (USD, EUR, etc.)
status	ENUM	Draft, Pending_Approval, Approved, Rejected, Active, Expired, Terminated
created_by, approved_by	INT (FK)	User references
is_renewable	BOOLEAN	Can be renewed
auto_renew	BOOLEAN	Automatic renewal
renewal_notice_days	INT	Days before renewal notice
is_deleted	BOOLEAN	Soft delete flag

Status Workflow:

- 1. **Draft:** Initial creation
- 2. **Pending Approval:** Submitted for review
- 3. **Approved/Rejected:** Approval decision
- 4. **Active:** Currently in effect
- 5. **Expired:** Past end date
- 6. **Terminated:** Manually ended

ContractRenewals Table

Purpose: Manages contract renewal requests and approvals

Column	Type	Description
renewal_id	INT (PK)	Unique renewal identifier
contract_id	INT (FK)	Parent contract
renewal_number	INT	Sequential renewal number
previous_end_date	DATE	Original end date
new_end_date	DATE	Proposed new end date
renewal_terms	TEXT	Updated terms
new_contract_value	DECIMAL(15,2)	Updated value
status	ENUM	Pending, Approved, Rejected
requested_by, approved_by	INT (FK)	User references

AuditTrail Table

Purpose: Complete audit log of all contract activities

Column	Type	Description
audit_id	INT (PK)	Unique audit record
contract_id	INT (FK)	Related contract
action_type	ENUM	Created, Updated, Approved, Rejected, Renewed, Terminated, Deleted, Restored
old_values, new_values	JSON	Before/after values
performed_by	INT (FK)	User who performed action
performed_at	TIMESTAMP	When action occurred
ip_address	VARCHAR(45)	User's IP address
user_agent	TEXT	Browser information
notes	TEXT	Additional context

Documents Table

Purpose: Contract document storage and versioning

Column	Type	Description
document_id	INT (PK)	Unique document identifier
contract_id	INT (FK)	Parent contract
document_name	VARCHAR(255)	File name
document_type	ENUM	Contract, Amendment, Addendum, Certificate, Other
file_path, file_url	VARCHAR(500)	Storage location
file_size	INT	File size in bytes
mime_type	VARCHAR(100)	File type
version_number	INT	Document version
is_current_version	BOOLEAN	Latest version flag
uploaded_by	INT (FK)	User who uploaded

Notifications Table

Purpose: System notifications and alerts

Column	Type	Description
notification_id	INT (PK)	Unique notification ID
contract_id	INT (FK)	Related contract
user_id	INT (FK)	Recipient user
notification_type	ENUM	Expiry_Warning, Renewal_Due, Approval_Required, Status_Change, Document_Upload
title	VARCHAR(200)	Notification title
message	TEXT	Notification content
priority	ENUM	Low, Medium, High, Critical
is_read	BOOLEAN	Read status
scheduled_at	TIMESTAMP	When to send
sent_at	TIMESTAMP	When actually sent

ContractAmendments Table

Purpose: Contract modification tracking

Column	Type	Description
amendment_id	INT (PK)	Unique amendment ID
contract_id	INT (FK)	Parent contract
amendment_number	INT	Sequential amendment number
amendment_title	VARCHAR(200)	Amendment title
amendment_description	TEXT	What changed
amendment_type	ENUM	Value_Change, Date_Extension, Scope_Change, Terms_Change, Other
original_value, amended_value	JSON	Before/after values
effective_date	DATE	When amendment takes effect
status	ENUM	Draft, Pending_Approval, Approved, Rejected
created_by, approved_by	INT (FK)	User references

Business Logic

Contract Approval Workflow

1. **Creation:** User creates contract in Draft status
2. **Submission:** Status changes to Pending_Approval
3. **Review:** Approver reviews and decides
4. **Decision:** Status becomes Approved/Rejected
5. **Activation:** Approved contracts become Active on start_date

Renewal Process

1. **Notification:** System generates renewal reminders based on `renewal_notice_days`
2. **Request:** User creates renewal request
3. **Approval:** Renewal follows approval workflow
4. **Execution:** Approved renewals update contract end_date

Soft Delete Implementation

- Contracts are never physically deleted
- `is_deleted` flag marks contracts as deleted
- Trigger prevents hard deletes
- Audit trail records deletion/restoration

Data Integrity Rules

- End date must be after start date
- Contract value must be non-negative
- Renewal notice days must be non-negative
- File path or URL required for documents
- Amendment effective dates validated

Views

ActiveContracts View

Provides easy access to currently active contracts with user information:

```
sql

SELECT contract_id, contract_number, title, status,
        created_by_name, approved_by_name
FROM ActiveContracts;
```

ContractsExpiringSoon View

Lists contracts expiring within 30 days:

```
sql

SELECT contract_number, title, end_date, days_until_expiry
FROM ContractsExpiringSoon
ORDER BY days_until_expiry;
```

Stored Procedures

ApproveContract

Purpose: Approve or reject a contract with proper validation

```
sql

CALL ApproveContract(contract_id, approver_id, 'APPROVE', 'Approved with conditions');
```

Parameters:

- `p_contract_id`: Contract to approve/reject
- `p_approver_id`: User performing action
- `p_action`: 'APPROVE' or 'REJECT'
- `p_notes`: Optional approval notes

SoftDeleteContract

Purpose: Safely delete contracts with audit trail

sql

```
CALL SoftDeleteContract(contract_id, user_id, 'Contract no longer needed');
```

GenerateExpiryNotifications

Purpose: Create notifications for expiring contracts

sql

```
CALL GenerateExpiryNotifications();
```

Run daily via scheduled job

Triggers

prevent_contract_hard_delete

- Prevents accidental data loss
- Forces use of soft delete pattern

contract_audit_trigger

- Automatically logs all contract updates
- Captures before/after values in JSON format
- Generates contextual notes

contract_creation_audit

- Logs contract creation events
- Records initial contract values

Usage Examples

Creating a New Contract

sql

```
INSERT INTO Contracts (  
    contract_number, title, description, contract_type,  
    party_a, party_b, start_date, end_date,  
    contract_value, currency, created_by  
) VALUES (  
    'CTR-2024-001', 'Software License Agreement',  
    'Annual software licensing contract', 'Service',  
    'Our Company', 'Software Vendor', '2024-01-01', '2024-12-31',  
    50000.00, 'USD', 1  
);
```

Uploading Contract Documents

sql

```
INSERT INTO Documents (  
    contract_id, document_name, document_type,  
    file_path, file_size, mime_type, uploaded_by  
) VALUES (  
    1, 'signed_contract.pdf', 'Contract',  
    '/uploads/contracts/2024/signed_contract.pdf',  
    2048576, 'application/pdf', 1  
);
```

Querying Contract History

sql

```
SELECT at.action_type, at.performed_at,  
    CONCAT(u.first_name, ' ', u.last_name) as performed_by,  
    at.notes  
FROM AuditTrail at  
JOIN Users u ON at.performed_by = u.user_id  
WHERE at.contract_id = 1  
ORDER BY at.performed_at DESC;
```

Finding Contracts Needing Renewal

sql

```
SELECT c.contract_number, c.title, c.end_date,  
       DATEDIFF(c.end_date, CURDATE()) as days_remaining  
FROM Contracts c  
WHERE c.is_renewable = TRUE  
      AND c.status = 'Active'  
      AND c.end_date <= DATE_ADD(CURDATE(), INTERVAL c.renewal_notice_days DAY)  
      AND c.is_deleted = FALSE;
```

Generating Monthly Contract Report

sql

```
SELECT  
    contract_type,  
    COUNT(*) as total_contracts,  
    SUM(contract_value) as total_value,  
    AVG(contract_value) as avg_value  
FROM Contracts  
WHERE status = 'Active'  
      AND is_deleted = FALSE  
GROUP BY contract_type  
ORDER BY total_value DESC;
```

Security Considerations

Data Protection

- Password hashing required for user accounts
- Sensitive audit information (IP, user agent) captured
- Soft delete prevents accidental data loss

Access Control

- Role-based permissions enforced at application level
- Foreign key constraints maintain referential integrity
- Audit trail provides accountability

Data Validation

- Check constraints prevent invalid data
- ENUM types restrict status values
- Date validation ensures logical consistency

Performance Optimization

Indexing Strategy

- Primary keys on all tables
- Foreign key indexes for join performance
- Composite indexes for common query patterns
- Functional indexes on computed columns

Key Indexes

```
sql

-- Contract queries
idx_contracts_status_dates: (status, end_date, is_deleted)
idx_contracts_created_by_status: (created_by, status, is_deleted)

-- Audit queries
idx_audit_contract_action_date: (contract_id, action_type, performed_at)

-- Notification queries
idx_notifications_user_read: (user_id, is_read, scheduled_at)
```

Query Optimization Tips

1. Always include `(is_deleted = FALSE)` in contract queries
2. Use date ranges efficiently with proper indexes
3. Limit audit trail queries with date boundaries
4. Consider pagination for large result sets

Maintenance Tasks

- **Daily:** Run `(GenerateExpiryNotifications())`
- **Weekly:** Archive old audit records
- **Monthly:** Analyze query performance
- **Quarterly:** Review and optimize indexes

Integration Guidelines

Application Integration

1. Always use stored procedures for complex operations
2. Implement proper error handling for constraint violations
3. Use transactions for multi-table operations

4. Validate data at both application and database levels

Reporting Integration

1. Use views for standard reports
2. Create materialized views for complex aggregations
3. Implement proper date filtering for performance
4. Consider read replicas for heavy reporting workloads

Backup and Recovery

1. Regular full database backups
2. Transaction log backups for point-in-time recovery
3. Test restore procedures regularly
4. Document recovery procedures

Version: 1.0

Last Updated: June 2025

Database Engine: MySQL 8.0+

Character Set: UTF8MB4