FluentCart Wishcart Database Structure Design

Executive Summary

Based on comprehensive analysis of leading WooCommerce wishlist plugins (YITH Wishlist with 900,000+ installations, TI Wishlist, WooCommerce Wishlists), I've designed a robust database structure for FluentCart's Wishcart feature. The design supports multiple wishlists per user, privacy controls, sharing functionality, guest support, and analytics tracking.

Recommended Database Tables: 7 Core Tables

---

1. Primary Tables Structure

1.1 `fc_wishlists` (Main Wishlist Table)

Stores the main wishlist records with metadata.

sql

```sql
CREATE TABLE fc_wishlists (
    id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    wishlist_token VARCHAR(64) NOT NULL UNIQUE,
    user_id BIGINT(20) UNSIGNED NULL DEFAULT NULL,
    session_id VARCHAR(255) NULL DEFAULT NULL,
    wishlist_name VARCHAR(255) NOT NULL DEFAULT 'My Wishlist',
    wishlist_slug VARCHAR(255) NOT NULL,
    description TEXT NULL,
    privacy_status ENUM('public', 'shared', 'private') DEFAULT 'private',
    is_default TINYINT(1) DEFAULT 0,
    expiration_date DATETIME NULL DEFAULT NULL,
    dateadded DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    date_modified DATETIME NULL ON UPDATE CURRENT_TIMESTAMP,
    menu_order INT(11) NOT NULL DEFAULT 0,
    wishlist_type VARCHAR(50) DEFAULT 'wishlist',
    status VARCHAR(20) DEFAULT 'active',
    PRIMARY KEY (id),
    KEY user_id_idx (user_id),
    KEY session_id_idx (session_id),
    KEY wishlist_token_idx (wishlist_token),
    KEY privacy_status_idx (privacy_status),
    KEY is_default_idx (is_default),
    KEY status_idx (status),
    KEY wishlist_slug_idx (wishlist_slug)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

1.2 `fc_wishlist_items` (Products in Wishlists)

Stores individual products added to wishlists.

sql

```sql
CREATE TABLE fc_wishlist_items (
    item_id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    wishlist_id BIGINT(20) UNSIGNED NOT NULL,
    product_id BIGINT(20) UNSIGNED NOT NULL,
    variation_id BIGINT(20) UNSIGNED NULL DEFAULT 0,
    variation_data LONGTEXT NULL,
    quantity INT(11) NOT NULL DEFAULT 1,
    date_added DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    position INT(11) DEFAULT 0,
    original_price DECIMAL(19, 4) NULL,
    original_currency VARCHAR(10) NULL,
    on_sale TINYINT(1) DEFAULT 0,
    notes TEXT NULL,
    user_id BIGINT(20) UNSIGNED NULL,
    date_added_to_cart DATETIME NULL,
    cart_item_key VARCHAR(255) NULL,
    custom_attributes TEXT NULL,
    status VARCHAR(20) DEFAULT 'active',
    PRIMARY KEY (item_id),
    UNIQUE KEY wishlist_product_unique (wishlist_id, product_id, variation_id),
    KEY wishlist_id_idx (wishlist_id),
    KEY product_id_idx (product_id),
    KEY variation_id_idx (variation_id),
    KEY date_added_idx (date_added),
    KEY user_id_idx (user_id),
    KEY status_idx (status),
    CONSTRAINT fk_wishlist_items_wishlist
        FOREIGN KEY (wishlist_id) REFERENCES fc_wishlists(id)
        ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

1.3 `fc_wishlist_shares` (Sharing & Social Media)

Tracks wishlist sharing activities and social media shares.

sql

```sql
CREATE TABLE fc_wishlist_shares (
    share_id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    wishlist_id BIGINT(20) UNSIGNED NOT NULL,
    share_token VARCHAR(64) UNIQUE,
    share_type ENUM('link', 'email', 'facebook', 'twitter', 'pinterest', 'whatsapp', 'instagram', 'other') NOT NULL,
    shared_by_user_id BIGINT(20) UNSIGNED NULL,
    shared_with_email VARCHAR(255) NULL,
    share_key VARCHAR(255) NULL,
    share_title VARCHAR(255) NULL,
    share_message TEXT NULL,
    click_count INT(11) DEFAULT 0,
    conversion_count INT(11) DEFAULT 0,
    date_created DATETIME DEFAULT CURRENT_TIMESTAMP,
    date_expires DATETIME NULL,
    last_clicked DATETIME NULL,
    status VARCHAR(20) DEFAULT 'active',
    PRIMARY KEY (share_id),
    KEY wishlist_id_idx (wishlist_id),
    KEY share_token_idx (share_token),
    KEY share_type_idx (share_type),
    KEY shared_by_user_idx (shared_by_user_id),
    KEY status_idx (status),
    CONSTRAINT fk_shares_wishlist
        FOREIGN KEY (wishlist_id) REFERENCES fc_wishlists(id)
        ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

1.4 `fc_wishlist_analytics` (Analytics & Popular Products)

Tracks wishlist analytics and popular products.

```sql
```

```sql
CREATE TABLE fc_wishlist_analytics (
    analytics_id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    product_id BIGINT(20) UNSIGNED NOT NULL,
    variation_id BIGINT(20) UNSIGNED NULL DEFAULT 0,
    wishlist_count INT(11) DEFAULT 0,
    click_count INT(11) DEFAULT 0,
    add_to_cart_count INT(11) DEFAULT 0,
    purchase_count INT(11) DEFAULT 0,
    share_count INT(11) DEFAULT 0,
    first_added_date DATETIME NULL,
    last_added_date DATETIME NULL,
    last_purchased_date DATETIME NULL,
    average_days_in_wishlist DECIMAL(10, 2) DEFAULT 0,
    conversion_rate DECIMAL(5, 2) DEFAULT 0,
    date_updated DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (analytics_id),
    UNIQUE KEY product_variation_unique (product_id, variation_id),
    KEY product_id_idx (product_id),
    KEY wishlist_count_idx (wishlist_count),
    KEY conversion_rate_idx (conversion_rate)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

---

2. Supporting Tables

2.1 `fc_wishlist_notifications` (Email & Notification Queue)

Manages email notifications and promotional campaigns.

```sql
```

```sql
CREATE TABLE fc_wishlist_notifications (
    notification_id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    user_id BIGINT(20) UNSIGNED NULL,
    wishlist_id BIGINT(20) UNSIGNED NULL,
    product_id BIGINT(20) UNSIGNED NULL,
    notification_type ENUM('price_drop', 'back_in_stock', 'promotional', 'reminder', 'share_notification', 'estimate_request') NO
    email_to VARCHAR(255) NOT NULL,
    email_subject VARCHAR(255) NULL,
    email_content LONGTEXT NULL,
    trigger_data TEXT NULL,
    scheduled_date DATETIME NULL,
    sent_date DATETIME NULL,
    opened_date DATETIME NULL,
    clicked_date DATETIME NULL,
    status ENUM('pending', 'sent', 'failed', 'cancelled') DEFAULT 'pending',
    attempts INT(3) DEFAULT 0,
    error_message TEXT NULL,
    date_created DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (notification_id),
    KEY user_id_idx (user_id),
    KEY wishlist_id_idx (wishlist_id),
    KEY product_id_idx (product_id),
    KEY notification_type_idx (notification_type),
    KEY status_idx (status),
    KEY scheduled_date_idx (scheduled_date)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

2.2 `fc_wishlist_activities` (Activity Log)

Tracks all wishlist activities for audit and user history.

```sql
```

```sql
CREATE TABLE fc_wishlist_activities (
    activity_id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    wishlist_id BIGINT(20) UNSIGNED NULL,
    user_id BIGINT(20) UNSIGNED NULL,
    session_id VARCHAR(255) NULL,
    activity_type ENUM('created', 'added_item', 'removed_item', 'moved_item', 'shared', 'viewed', 'renamed', 'deleted', 'purchase
    object_id BIGINT(20) UNSIGNED NULL,
    object_type VARCHAR(50) NULL,
    activity_data TEXT NULL,
    ip_address VARCHAR(45) NULL,
    user_agent TEXT NULL,
    referrer_url TEXT NULL,
    date_created DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (activity_id),
    KEY wishlist_id_idx (wishlist_id),
    KEY user_id_idx (user_id),
    KEY activity_type_idx (activity_type),
    KEY date_created_idx (date_created)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

2.3 `fc_wishlist_guest_users` (Guest User Management)

Manages guest user sessions and data.

sql

```sql
CREATE TABLE fc_wishlist_guest_users (
    guest_id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
    session_id VARCHAR(255) NOT NULL UNIQUE,
    guest_email VARCHAR(255) NULL,
    guest_name VARCHAR(255) NULL,
    ip_address VARCHAR(45) NULL,
    user_agent TEXT NULL,
    wishlist_data LONGTEXT NULL,
    conversion_user_id BIGINT(20) UNSIGNED NULL,
    date_created DATETIME DEFAULT CURRENT_TIMESTAMP,
    date_expires DATETIME NULL,
    last_activity DATETIME NULL,
    PRIMARY KEY (guest_id),
    KEY session_id_idx (session_id),
    KEY guest_email_idx (guest_email),
    KEY date_expires_idx (date_expires),
    KEY conversion_user_id_idx (conversion_user_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

---

3. Database Design Rationale

Key Design Decisions:

1. **Separate Tables for Scalability**
   - Main wishlist table separate from items for better query performance
   - Analytics separated to avoid constant updates to main tables
   - Activity logging in dedicated table for audit trails
2. **Support for Multiple Wishlists**
   - Each user can create unlimited wishlists
   - `is_default` flag for primary wishlist
   - `wishlist_type` for future extensions (registry, favorites, etc.)
3. **Privacy & Sharing Architecture**
   - Three privacy levels: public, shared, private
   - Separate shares table for tracking social media and link shares
   - Token-based sharing for security
4. **Guest User Support**
   - Session-based wishlists for non-registered users
   - Automatic conversion when guest registers
   - Configurable expiration for guest wishlists
5. **Analytics & Insights**
   - Dedicated analytics table for performance
   - Popular products tracking
   - Conversion rate calculations
   - Activity logging for user behavior analysis
6. **Variation Support**
   - Full support for product variations
   - JSON storage for variation data
   - Custom attributes support

---

4. Index Strategy

Primary Indexes:

- User lookups: `user_id` on all relevant tables

- Product queries: `product_id` and `variation_id`
- Token searches: `wishlist_token`, `share_token`
- Performance queries: Composite indexes for common JOIN operations

Recommended Additional Indexes:

```sql
-- Composite indexes for common queries
CREATE INDEX idx_wishlist_user_status ON fc_wishlists(user_id, status);
CREATE INDEX idx_items_product_date ON fc_wishlist_items(product_id, date_added);
CREATE INDEX idx_analytics_popular ON fc_wishlist_analytics(wishlist_count DESC, product_id);
CREATE INDEX idx_notifications_pending ON fc_wishlist_notifications(status, scheduled_date)
    WHERE status = 'pending';
```

---

5. Migration & Upgrade Path

From Basic to Advanced Features:

Phase 1 (MVP):

- Tables: `fc_wishlists`, `fc_wishlist_items`
- Basic wishlist functionality

Phase 2 (Social Features):

- Add: `fc_wishlist_shares`
- Enable sharing and social media integration

Phase 3 (Analytics):

- Add: `fc_wishlist_analytics`, `fc_wishlist_activities`
- Enable insights and tracking

Phase 4 (Advanced):

- Add: `fc_wishlist_notifications`, `fc_wishlist_guest_users`
- Full feature parity with premium WooCommerce plugins

---

6. Performance Considerations

Optimization Strategies:

1. **Caching Layer:**
    - Redis/Memcached for frequently accessed wishlists
    - Cache popular products data

- Session-based caching for guest users

2. **Query Optimization:**
   - Prepared statements for all queries
   - Batch inserts for bulk operations
   - Lazy loading for wishlist items
3. **Maintenance:**
   - Automated cleanup of expired guest wishlists
   - Archive old activity logs
   - Periodic analytics aggregation
4. **Scalability:**
   - Horizontal partitioning by user_id if needed
   - Read replicas for analytics queries
   - Asynchronous processing for notifications

---

7. Security Measures

1. **Data Protection:**
   - Encryption for sensitive data
   - Token generation using cryptographically secure methods
   - SQL injection prevention through prepared statements
2. **Privacy Compliance:**
   - GDPR-compliant data structure
   - User data export capability
   - Right to deletion support
3. **Access Control:**
   - Row-level security for wishlists
   - Token validation for shared wishlists
   - Rate limiting for API endpoints

---

8. API Integration Points

RESTful Endpoints:

```
GET    /api/wishlists            # List user's wishlists
POST   /api/wishlists            # Create new wishlist
GET    /api/wishlists/{token}    # Get specific wishlist
PUT    /api/wishlists/{token}    # Update wishlist
DELETE /api/wishlists/{token}    # Delete wishlist

POST   /api/wishlists/{token}/items     # Add item
DELETE /api/wishlists/{token}/items/{id} # Remove item
POST   /api/wishlists/{token}/share     # Share wishlist
GET    /api/wishlists/popular           # Popular products
```

---

9. Comparison with WooCommerce Plugins

Feature Coverage:

Feature	YITH	TI Wishlist	Our Design
Multiple Wishlists	✓	✓	✓
Privacy Controls	✓	✓	✓
Social Sharing	✓	✓	✓
Guest Support	✓	✓	✓
Analytics	Premium	Premium	✓
Email Notifications	Premium	Premium	✓
Activity Tracking	Premium	-	✓
Variation Support	✓	✓	✓
API Access	-	-	✓
Performance Optimized	-	-	✓

---

10. Implementation Recommendations

Priority Order:

1. **Core Tables First:**
   - Start with `fc_wishlists` and `fc_wishlist_items`
   - Implement basic CRUD operations
   - Add frontend integration
2. **Enhanced Features:**
   - Add sharing capabilities with `fc_wishlist_shares`

- Implement guest user support
- Enable multiple wishlists

3. **Advanced Features:**
   - Analytics and reporting
   - Email notifications
   - Activity tracking
   - API development

Development Timeline:

- **Week 1-2:** Core tables and basic functionality
- **Week 3-4:** Sharing and privacy features
- **Week 5-6:** Analytics and notifications
- **Week 7-8:** Testing, optimization, and launch

Conclusion

This 7-table database structure provides FluentCart with a robust, scalable foundation for the Wishcart feature that matches and exceeds the capabilities of leading WooCommerce wishlist plugins. The design supports all essential features while maintaining performance and allowing for future expansion.

The modular approach allows for phased implementation, starting with core functionality and progressively adding advanced features based on user feedback and business priorities.