

Database Attribute Analysis Summary

Date: 2025-10-13

Migration: 2025_10_13_162211_remove_unused_columns_from_tables.php

Removed Unused Attributes

The following attributes were identified as unused and removed from the database:

1. users.normalized_name ✗

- **Reason:** The old `Employee` model had a boot method that populated this field, but after renaming to `User` model, this logic was never migrated
- **Impact:** No code references this column
- **Status:** Removed

2. users.payroll_day ✗

- **Type:** `smallint`
- **Reason:** Payroll features were never implemented
- **Impact:** Column exists but is never set or read
- **Status:** Removed

3. users.salary ✗

- **Type:** `numeric(10,2)`
- **Reason:** Payroll features were never implemented
- **Impact:** Column exists but is never set or read
- **Status:** Removed

4. leave_requests.is_seen ✗

- **Type:** `boolean`
- **Reason:** Notification/seen tracking was planned but never implemented
- **Impact:** Column exists with default `false`, but no code sets or reads it
- **Status:** Removed

5. shiftschedules.start_time ✗

- **Type:** `time`
- **Reason:** UI hardcodes shift times (morning: 06:00-15:00, evening: 15:00-00:00); backend never uses these columns
- **Impact:** Columns exist but are never populated or referenced in logic
- **Status:** Removed

6. shiftschedules.end_time ✗

- **Type:** `time`
 - **Reason:** UI hardcodes shift times; backend never uses these columns
 - **Impact:** Columns exist but are never populated or referenced in logic
 - **Status:** Removed
-

Retained Attributes (With Recommendations)

Leave Balances (Currently Unused, Keep for Future)

- users.annual_leave_balance (int) ⚠
- users.sick_leave_balance (int) ⚠
- users.emergency_leave_balance (int) ⚠

Status: Kept

Usage: Only reconciliation migrations update these values; no enforcement in leave approval workflow

Recommendation: Implement balance checking when approving leave requests to prevent negative balances

All Remaining Attributes (By Table)

users (17 attributes - all in use)

1. id - Primary key
2. name - Display name
3. email - Login, unique
4. phone - Contact, unique
5. national_id - Government ID, unique
6. password - Authentication
7. email_verified_at - Email verification
8. remember_token - Session persistence
9. userRole - Authorization (owner/admin/staff)
10. annual_leave_balance - Leave tracking (future enforcement)
11. sick_leave_balance - Leave tracking (future enforcement)
12. emergency_leave_balance - Leave tracking (future enforcement)
13. hired_on - Employment start date
14. address - HR records
15. bank_acc_no - Payroll (future)
16. deleted_at - Soft delete
17. created_at, updated_at - Audit trail

attendances (12 attributes - all in use)

1. id - Primary key
2. user_id - FK to users
3. schedule_id - FK to shiftschedules (nullable)
4. date - Attendance date
5. status - on_time/late/missed/present
6. sign_in_time - Clock in
7. sign_off_time - Clock out
8. notes - Admin notes
9. is_manually_filled - Entry method
10. deleted_at - Soft delete
11. created_at, updated_at - Audit trail

leave_requests (11 attributes - all in use)

1. id - Primary key
2. user_id - FK to users
3. type - Annual/Sick/Emergency

4. `start_date` - Leave start
5. `end_date` - Leave end (nullable)
6. `message` - Employee reason
7. `status` - 0=Pending, 1=Approved, 2=Rejected
8. `admin_response` - Admin feedback
9. `deleted_at` - Soft delete
10. `created_at`, `updated_at` - Audit trail

shiftschedules (10 attributes - all in use)

1. `id` - Primary key
2. `user_id` - FK to users
3. `shift_type` - morning/evening
4. `week_start` - Week identifier
5. `day` - Specific date
6. `submitted` - Week lock status
7. `deleted_at` - Soft delete
8. `created_at`, `updated_at` - Audit trail

personal_access_tokens (10 attributes - Laravel Sanctum)

All attributes are framework-managed (Sanctum).

password_reset_tokens (3 attributes - Laravel Auth)

All attributes are framework-managed (Auth).

Documentation Updates

1. **ERD.mmd (Entity-Relationship Diagram)**

- Removed: `normalized_name`, `payroll_day`, `salary`, `is_seen`, `start_time`, `end_time`
- Added: Complete attribute list with all remaining columns
- Updated: Relationships to show `shiftschedules` → `attendances` link

2. **DomainClassDiagram.mmd (Class Diagram)**

- Removed: Same as ERD
- Added: Complete attribute list with types and constraints
- Updated: Model methods to include both `user()` and `employee()` for backward compatibility

3. **Relationships.md (Comprehensive Documentation)**

- **New Section:** "Comprehensive Attribute Details" with full table breakdown:
 - Attribute name
 - Data type
 - Constraints
 - Domain values
 - Objective
 - Usage in code
- **Updated Section:** "Integrity Constraints & Validations"
- **Updated Section:** "Indexes & Performance"
- **Added:** Summary of changes from initial state

4. PDFs Generated

- `ERD.pdf` - Visual entity-relationship diagram
 - `DomainClassDiagram.pdf` - Visual class diagram
-

Database State After Cleanup

Tables: 6

1. `users` (17 cols) - Core user/employee data
2. `attendances` (12 cols) - Daily attendance tracking
3. `leave_requests` (11 cols) - Leave applications
4. `shiftschedules` (10 cols) - Weekly shift assignments
5. `personal_access_tokens` (10 cols) - API auth (Sanctum)
6. `password_reset_tokens` (3 cols) - Password resets (Auth)

Total Attributes: 63

- **Active/Used:** 63 (100%)
- **Removed/Cleanup:** 6 (normalized_name, payroll_day, salary, is_seen, start_time, end_time)

Data Integrity

- All foreign keys valid
 - All unique constraints enforced
 - All check constraints active
 - Soft deletes on core tables
 - No orphaned columns
 - No dead code references
-

Next Steps (Optional Enhancements)

1. Implement Leave Balance Enforcement:

- Check balances before approving leave requests
- Surface balances in UI for staff visibility
- Add validation to prevent over-requesting

2. Enhanced Shift Time Handling (if needed in future):

- If you want flexible shift times (not hardcoded), re-add `start_time` / `end_time` to `shiftschedules`
- Use these for lateness calculation instead of hardcoded values
- Populate during schedule creation

3. Notification System (if needed in future):

- If you want "seen" tracking for leave requests, re-add `is_seen` column
- Implement notification read/unread logic

4. Payroll System (if needed in future):

- If you plan payroll, re-add `payroll_day` and `salary` to `users`
 - Implement payroll calculation based on attendance and salary
-

Status:  Database is now clean, documented, and ready for production use.