

Database Relationships & Comprehensive Attribute Analysis

Table of Contents

1. [Entities & Primary Keys](#)
 2. [Relationships \(Cardinality + Verbs\)](#)
 3. [Comprehensive Attribute Details](#)
 4. [Integrity Constraints & Validations](#)
 5. [Indexes & Performance](#)
-

Entities & Primary Keys

- **users** (PK: id)
 - **attendances** (PK: id, FK: user_id → users.id, FK: schedule_id → shiftschedules.id nullable)
 - **leave_requests** (PK: id, FK: user_id → users.id)
 - **shiftschedules** (PK: id, FK: user_id → users.id)
 - **personal_access_tokens** (PK: id, polymorphic tokenable)
 - **password_reset_tokens** (PK: email)
-

Relationships (Cardinality + Verbs)

users 1 — M attendances

- **Verb:** User **HAS** many attendances; Attendance **BELONGS TO** one user
- **FK:** attendances.user_id → users.id (ON DELETE CASCADE)
- **Optional FK:** attendances.schedule_id → shiftschedules.id (ON DELETE SET NULL)
- **Unique:** (user_id, date) ensures at most one attendance per user per day
- **Indexes:**
 - (user_id, date) - for daily lookups
 - (user_id, date, status) - for reporting
 - Partial index on status = 'present' - for present-only queries
- **Purpose:** Track daily attendance records (clock in/out times, status)

users 1 — M leave_requests

- **Verb:** User **HAS** many leave requests; Leave request **BELONGS TO** one user
- **FK:** leave_requests.user_id → users.id (ON DELETE CASCADE)
- **Constraints:**
 - status ∈ {0, 1, 2} (0=Pending, 1=Approved, 2=Rejected)
 - end_date ≥ start_date
- **Indexes:**
 - user_id - for user's leave history
 - status - for filtering by approval state
 - type - for filtering by leave type
 - (user_id, status) - composite for user's pending/approved leaves
 - Partial index on status = 0 - for pending requests dashboard
- **Purpose:** Manage leave applications (annual, sick, emergency)

users 1 — M shiftschedules

- **Verb:** User **HAS** many shift schedules; Shift schedule **BELONGS TO** one user
- **FK:** `shiftschedules.user_id` → `users.id` (ON DELETE CASCADE)
- **Unique:** (`week_start`, `day`, `shift_type`) enforces one assignee per shift slot per day
- **Indexes:**
 - `(user_id, day)` - for user's daily schedule
 - `(user_id, week_start)` - for user's weekly schedule
- **Purpose:** Assign staff to specific shift slots (morning/evening) per day

shiftschedules 1 — M attendances (optional link)

- **Verb:** Schedule **HAS** many attendances; Attendance **BELONGS TO** one schedule
- **FK:** `attendances.schedule_id` → `shiftschedules.id` (nullable, ON DELETE SET NULL)
- **Usage:** Links attendance to specific shift slot; enables shift-specific reporting
- **Purpose:** Track which shift schedule an attendance was for; enforce clock-in only when scheduled

users 1 — M personal_access_tokens (polymorphic)

- **Verb:** User **OWNS** many tokens; Token **BELONGS TO** a tokenable (user)
- **Polymorphic:** `tokenable_type/tokenable_id` pair points to users
- **Purpose:** API authentication via Laravel Sanctum

Comprehensive Attribute Details

Table: users

Domain: Staff/Employee records **Objective:** Central registry of all system users (staff members)

| Attribute | Type | Constraint | Domain Values | Objective | Usage |
|--------------------------|--------------|--------------------|--------------------|----------------------------|-------------------------------------|
| <code>id</code> | bigint | PK, auto-increment | 1..∞ | Unique identifier | Primary key for all user operations |
| <code>name</code> | varchar(255) | NOT NULL | Any valid name | Employee's full name | Display in user reports, schedules |
| <code>email</code> | varchar(255) | UNIQUE, NOT NULL | Valid email format | Login credential & contact | Authentication, notification |
| <code>phone</code> | varchar(255) | UNIQUE, NOT NULL | Phone number | Contact information | Uniqueness validation, contact |
| <code>national_id</code> | varchar(255) | UNIQUE, NOT NULL | National ID number | Government ID | Uniqueness validation, records |
| <code>password</code> | varchar(255) | NOT NULL | Hashed password | Authentication credential | Login security (bcrypt) |

| | | | | | |
|-------------------------|--------------|---------------------------|---------------------------|--------------------------------|---|
| | | | | | hashed) |
| email_verified_at | timestamp | NULL | Timestamp or NULL | Email verification status | Laravel auto email verification |
| remember_token | varchar(100) | NULL | Token or NULL | Session persistence | "Remember me" login feature |
| userRole | varchar(255) | NOT NULL, default 'staff' | 'owner', 'admin', 'staff' | Authorization level | Role-based access control (replaced Spatie) |
| annual_leave_balance | int | >= 0, default 15 | 0..∞ | Remaining annual leave days | Leave approval validation (future) |
| sick_leave_balance | int | >= 0, default 10 | 0..∞ | Remaining sick leave days | Leave approval validation (future) |
| emergency_leave_balance | int | >= 0, default 5 | 0..∞ | Remaining emergency leave days | Leave approval validation (future) |
| hired_on | date | NULL | Valid date | Employment start date | Tenure calculation: reports |
| address | text | NULL | Any address | Employee's address | HR records: payroll |
| bank_acc_no | varchar(255) | NULL | Bank account number | Payroll deposit account | Salary payments (future) |
| deleted_at | timestamp | NULL | Timestamp or NULL | Soft delete marker | Data recovery: audit trail |
| created_at | timestamp | NOT NULL | Timestamp | Record creation time | Audit trail |
| updated_at | timestamp | NOT NULL | Timestamp | Last modification time | Audit trail |

Removed Attributes: normalized_name (unused), payroll_day , salary (unused, removed in migration 2025_10_13_162211)

Table: attendances

Domain: Daily attendance records **Objective:** Track employee clock-in/out times and attendance status

| Attribute | Type | Constraint | Domain Values | Objective | Usage |
|---------------------------------|---------------------------|--|---|------------------------|---------------------------------------|
| <code>id</code> | <code>bigint</code> | PK, auto-increment | <code>1..∞</code> | Unique identifier | Primary key for attendance records |
| <code>user_id</code> | <code>bigint</code> | <code>FK → users.id, NOT NULL</code> | Valid user ID | Employee being tracked | Links attendance to specific user |
| <code>schedule_id</code> | <code>bigint</code> | <code>FK → shiftschedules.id, NULL</code> | Valid schedule ID or NULL | Shift slot reference | Links attendance to assigned shift |
| <code>date</code> | <code>date</code> | <code>NOT NULL, UNIQUE with user_id</code> | Valid date | Attendance date | Which day the attendance is for |
| <code>status</code> | <code>varchar(255)</code> | <code>NOT NULL</code> | <code>'on_time', 'late', 'missed', 'present'</code> | Attendance status | Performance tracking, reports |
| <code>sign_in_time</code> | <code>time</code> | <code>NULL</code> | <code>HH:MM:SS</code> | Clock-in time | Calculate working hours, lateness |
| <code>sign_off_time</code> | <code>time</code> | <code>NULL</code> | <code>HH:MM:SS</code> | Clock-out time | Calculate working hours |
| <code>notes</code> | <code>varchar(255)</code> | <code>NULL</code> | Any text | Additional context | Admin notes, reasons for manual entry |
| <code>is_manually_filled</code> | <code>boolean</code> | <code>default false</code> | <code>true/false</code> | Entry method indicator | Distinguish manual vs self clock-in |
| <code>deleted_at</code> | <code>timestamp</code> | <code>NULL</code> | <code>Timestamp or NULL</code> | Soft delete marker | Data recovery, corrections |

| | | | | | |
|------------|-----------|----------|-----------|------------------------|-------------|
| created_at | timestamp | NOT NULL | Timestamp | Record creation time | Audit trail |
| updated_at | timestamp | NOT NULL | Timestamp | Last modification time | Audit trail |

Removed Attributes: None (all current attributes are used)

Table: leave_requests

Domain: Leave applications **Objective:** Manage employee leave requests (annual, sick, emergency)

| Attribute | Type | Constraint | Domain Values | Objective | Usage |
|----------------|--------------|---------------------------------|---|---------------------------|--|
| id | bigint | PK, auto-increment | 1..∞ | Unique identifier | Primary key for leave requests |
| user_id | bigint | FK → users.id, NOT NULL | Valid user ID | Employee requesting leave | Links request to specific user |
| type | varchar(255) | NOT NULL | 'Annual Leave', 'Sick Leave', 'Emergency Leave' | Leave category | Determines which balance to deduct from |
| start_date | date | NOT NULL, <= end_date | Valid date | Leave start date | Defines leave period start |
| end_date | date | NULL, >= start_date | Valid date or NULL | Leave end date | Defines leave period end (NULL = single day) |
| message | text | NULL | Any text | Employee's reason | Context for admin approval decision |
| status | smallint | NOT NULL, default 0, IN (0,1,2) | 0=Pending, 1=Approved, 2=Rejected | Approval state | Workflow tracking, notifications |
| admin_response | text | NULL | Any text | Admin's feedback | Communicate rejection reasons |
| deleted_at | timestamp | NULL | Timestamp or NULL | Soft delete marker | Data recovery, corrections |

| | | | | | |
|------------|-----------|----------|-----------|-------------------------|--------------------|
| created_at | timestamp | NOT NULL | Timestamp | Request submission time | SLA tracking |
| updated_at | timestamp | NOT NULL | Timestamp | Last status change time | Approval timestamp |

Removed Attributes: `is_seen` (unused, removed in migration 2025_10_13_162211)

Table: shiftschedules

Domain: Weekly shift assignments **Objective:** Assign employees to specific shift slots (morning/evening) each day

| Attribute | Type | Constraint | Domain Values | Objective | Usage |
|-------------------------|--------------|--|----------------------|--------------------------|--|
| <code>id</code> | bigint | PK, auto-increment | 1..∞ | Unique identifier | Primary key for schedule assignments |
| <code>user_id</code> | bigint | FK → users.id, NOT NULL | Valid user ID | Assigned employee | Who is working this shift |
| <code>shift_type</code> | varchar(255) | NOT NULL, UNIQUE with (week_start, day) | 'morning', 'evening' | Shift slot type | Determines shift timing (morning 06:00-15:00, evening 15:00-00:00) |
| <code>week_start</code> | date | NOT NULL, UNIQUE with (day, shift_type) | Monday date | Week identifier | Groups shifts into weeks |
| <code>day</code> | date | NOT NULL, UNIQUE with (week_start, shift_type) | Valid date | Specific day | Which day this shift is for |
| <code>submitted</code> | boolean | default false | true/false | Week finalization status | Locks schedule editing for the week |
| <code>deleted_at</code> | timestamp | NULL | Timestamp or NULL | Soft delete marker | Data recovery, corrections |
| <code>created_at</code> | timestamp | NOT NULL | Timestamp | Assignment creation time | Audit trail |
| <code>updated_at</code> | timestamp | NOT NULL | Timestamp | Last reassignment time | Audit trail |

Removed Attributes: `start_time`, `end_time` (unused, hardcoded in UI, removed in migration 2025_10_13_162211)

Table: personal_access_tokens (Laravel Sanctum)

Domain: API authentication tokens **Objective:** Manage API access tokens for authenticated users

| Attribute | Type | Constraint | Domain Values | Objective | Usage |
|-----------------------------|---------------------------|--------------------|--------------------|------------------------|-----------------------------------|
| <code>id</code> | <code>bigint</code> | PK, auto-increment | 1..∞ | Unique identifier | Primary key |
| <code>tokenable_type</code> | <code>varchar(255)</code> | NOT NULL | 'App\Models\User' | Polymorphic model type | Identifies which model owns token |
| <code>tokenable_id</code> | <code>bigint</code> | NOT NULL | Valid user ID | Owner ID | Which user owns this token |
| <code>name</code> | <code>varchar(255)</code> | NOT NULL | Any name | Token description | Identifies token purpose |
| <code>token</code> | <code>varchar(64)</code> | UNIQUE, NOT NULL | 64-char hash | Access credential | Authentication bearer token |
| <code>abilities</code> | <code>text</code> | NULL | JSON array or NULL | Token permissions | Scope-based access control |
| <code>last_used_at</code> | <code>timestamp</code> | NULL | Timestamp or NULL | Last usage time | Inactivity tracking |
| <code>expires_at</code> | <code>timestamp</code> | NULL | Timestamp or NULL | Expiration time | Token validity period |
| <code>created_at</code> | <code>timestamp</code> | NOT NULL | Timestamp | Token creation time | Audit trail |
| <code>updated_at</code> | <code>timestamp</code> | NOT NULL | Timestamp | Last update time | Audit trail |

Table: password_reset_tokens (Laravel Auth)

Domain: Password reset requests **Objective:** Manage password reset token lifecycle

| Attribute | Type | Constraint | Domain Values | Objective | Usage |
|--------------------|---------------------------|------------|---------------|--------------|--------------------------------|
| <code>email</code> | <code>varchar(255)</code> | PK | Valid email | User's email | Identifies who requested reset |

| | | | | | |
|------------|--------------|----------|--------------|-----------------------|-----------------------------------|
| token | varchar(255) | NOT NULL | Hashed token | Reset credential | Secure reset link validation |
| created_at | timestamp | NULL | Timestamp | Request creation time | Token expiry calculation (60 min) |

Integrity Constraints & Validations

Database-Level Constraints

1. CHECK Constraints:

- o `users.annual_leave_balance >= 0`
- o `users.sick_leave_balance >= 0`
- o `users.emergency_leave_balance >= 0`
- o `leave_requests.status IN (0, 1, 2)`
- o `leave_requests.end_date >= start_date` (when both present)

2. UNIQUE Constraints:

- o `users.email , users.phone , users.national_id`
- o `attendances(user_id, date)` - one attendance per user per day
- o `shiftschedules(week_start, day, shift_type)` - one assignee per shift slot
- o `personal_access_tokens.token`

3. Foreign Key Constraints:

- o `attendances.user_id → users.id` (ON DELETE CASCADE)
- o `attendances.schedule_id → shiftschedules.id` (ON DELETE SET NULL)
- o `leave_requests.user_id → users.id` (ON DELETE CASCADE)
- o `shiftschedules.user_id → users.id` (ON DELETE CASCADE)

4. Soft Deletes:

- o Enabled on: `users , attendances , leave_requests , shiftschedules`
- o Purpose: Data recovery, audit trail, historical reporting

Application-Level Validations

1. Schedule Assignment (ScheduleController):

- o Block assignment if employee has approved leave on that day
- o Prevent duplicate assignments via DB unique constraint

2. Attendance Creation (AttendanceServices):

- o Block if employee has approved leave on that date
- o Require a valid schedule for that date
- o Auto-set `schedule_id` when creating attendance

3. Leave Balance Reconciliation (Migration):

- o Count actual days of approved leave (not just request count)
- o Deduct from appropriate balance based on leave type

| | | | | | |
|------------|--------------|----------|--------------|-----------------------|-----------------------------------|
| token | varchar(255) | NOT NULL | Hashed token | Reset credential | Secure reset link validation |
| created_at | timestamp | NULL | Timestamp | Request creation time | Token expiry calculation (60 min) |

Indexes & Performance

Primary Indexes (Auto-created)

- `users.id`, `attendances.id`, `leave_requests.id`, `shiftschedules.id`,
`personal_access_tokens.id`

Unique Indexes (Enforce constraints)

- `users_email_unique`, `users_phone_unique`, `users_national_id_unique`
- `attendances_user_id_date_unique`
- `shiftschedules_week_start_day_shift_type_unique`
- `personal_access_tokens_token_unique`

Foreign Key Indexes (Join performance)

- `attendances_user_id_index`
- `attendances_schedule_id_index`
- `leave_requests_user_id_index`
- `shiftschedules_user_id_index`

Composite Indexes (Reporting queries)

- `attendances_user_id_date_status_index` - dashboard attendance reports
- `leave_requests_user_id_status_index` - user's pending/approved leaves
- `shiftschedules_user_id_day_index` - daily schedule lookup
- `shiftschedules_user_id_week_start_index` - weekly schedule view

Partial Indexes (Filtered queries)

- `leave_requests_status_pending_index` WHERE `status = 0` - pending requests dashboard
- `attendances_status_present_index` WHERE `status = 'present'` - present-only reports

Single-Column Indexes (Filter/sort operations)

- `leave_requests_status_index`, `leave_requests_type_index`
- `attendances_date_index`, `shiftschedules_day_index`,
`shiftschedules_week_start_index`

Summary of Changes from Initial State

Removed (Cleanup)

- `users.normalized_name` (orphaned logic)
- `users.payroll_day`, `users.salary` (unused)
- `leave_requests.is_seen` (never implemented)
- `shiftschedules.start_time`, `shiftschedules.end_time` (hardcoded in UI)

Added (Enhancement)

- `attendances.schedule_id` FK for explicit shift linkage
- Soft deletes on all core tables
- CHECK constraints for data integrity
- Composite and partial indexes for performance
- Unique constraint on `shiftschedules(week_start, day, shift_type)` for slot uniqueness

Fixed (Corrections)

- Renamed `employees` → `users` , `requests` → `leave_requests` , `schedules` → `shiftschedules`
 - Fixed leave balance reconciliation to count days, not requests
 - Dropped duplicate/legacy indexes and sequences
 - Added missing indexes on frequently queried columns
 - Application-level validations to prevent conflicts (leave vs schedule/attendance)
-

Last Updated: 2025-10-13 after migration

`2025_10_13_162211_remove_unused_columns_from_tables.php`