Technical Specifications - Detailed Data Model V3

1. Introduction

This document defines the detailed data model for the geometric maturity tracking application based on the current database schema implementation. It describes the main data entities, their attributes, data types, and the relationships between them. This model serves as the foundation for the application's database structure.

2. Conventions

- **Common Data Types:**
  - `bigint`: Primary keys and foreign keys
  - `text`: Variable-length text fields
  - `integer`: Whole numbers
  - `decimal(precision,scale)`: Numbers with decimal points
  - `boolean`: True/False values (default values specified)
  - `date`: Date format (YYYY-MM-DD)
  - `timestamp with time zone`: Date and time with timezone
  - `uuid`: Universally unique identifier for user references
- **Primary Keys:** Auto-incrementing `bigint` for all entities
- **Audit Fields:** All entities include `created_at` and `updated_at` timestamps
- **Soft Delete:** Most entities include `is_active` boolean field

3. Core Entity Definitions

3.1. Projects

```sql
```

```sql
CREATE TABLE "projects" (
  "id" bigint PRIMARY KEY,
  "project_name" text NOT NULL,
  "project_code" text,
  "project_description" text,
  "project_start_date" date,
  "project_end_date" date,
  "project_manager_user_id" uuid,
  "project_status" project_status_enum DEFAULT 'Active' NOT NULL,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Represents projects within which components and their maturity are tracked.

Key Features:

- Unique project names and codes
- Optional start/end dates with validation
- Project manager assignment
- Status tracking (Active/Archived/Closed)

Constraints:

- `check_project_dates`: Ensures end date is after start date

3.2. Component Classifications

sql

```sql
CREATE TABLE "component_classifications" (
  "id" bigint PRIMARY KEY,
  "classification_code" text NOT NULL,
  "classification_description" text NOT NULL,
  "is_active" boolean DEFAULT true NOT NULL,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Defines component classifications (e.g., F1, F2, E1).

Key Features:

- Unique classification codes
- Soft delete capability via `is_active`

- Standardized classification system

3.3. Drawings

```sql
CREATE TABLE "drawings" (
  "id" bigint PRIMARY KEY,
  "item_code" text NOT NULL,
  "drawing_version" text NOT NULL,
  "component_description" text NOT NULL,
  "component_classification_id" bigint NOT NULL,
  "total_standard_dimensions" integer NOT NULL,
  "total_spc_dimensions" integer NOT NULL,
  "food_contact" boolean DEFAULT false NOT NULL,
  "removable" boolean DEFAULT false NOT NULL,
  "drawing_attachment_path" text,
  "revision_notes" text,
  "is_active" boolean DEFAULT true NOT NULL,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Master data for component drawings and their versions.

Key Features:

- Unique combination of item code and version
- Dimensional requirements tracking
- Component characteristics (food contact, removable)
- File attachment support
- Revision history via notes

Constraints:

- `drawings_total_standard_dimensions_check`: Standard dimensions ≥ 0
- `drawings_total_spc_dimensions_check`: SPC dimensions ≥ 0

3.4. Mold Locations

```sql
```

```sql
CREATE TABLE "mold_locations" (
    "id" bigint PRIMARY KEY,
    "location_name" text NOT NULL,
    "location_description" text,
    "location_address" text,
    "is_active" boolean DEFAULT true NOT NULL,
    "created_at" timestamp with time zone DEFAULT now() NOT NULL,
    "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Defines physical locations where molds are stored/operated.

3.5. Molds

sql

```sql
CREATE TABLE "molds" (
    "id" bigint PRIMARY KEY,
    "mold_code" text NOT NULL,
    "mold_description" text,
    "total_cavities" integer NOT NULL,
    "mold_type" mold_type_enum NOT NULL,
    "mold_location_id" bigint NOT NULL,
    "is_active" boolean DEFAULT true NOT NULL,
    "purchase_date" date,
    "maintenance_notes" text,
    "created_at" timestamp with time zone DEFAULT now() NOT NULL,
    "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Master data for molds used to produce components.

Key Features:

- Unique mold codes
- Cavity count tracking
- Type classification (Prototype/Low Volume/Definitive)
- Location assignment
- Maintenance history

Constraints:

- `molds_total_cavities_check`: Total cavities > 0

3.6. Parent Components

sql

```sql
CREATE TABLE "parent_components" (
    "id" bigint PRIMARY KEY,
    "project_id" bigint NOT NULL,
    "drawing_id" bigint NOT NULL,
    "mold_id" bigint NOT NULL,
    "calculated_parent_status" parent_status_enum,
    "last_status_calculation_date" timestamp with time zone,
    "planned_ot_date" date,
    "actual_ot_date" date,
    "planned_otop_date" date,
    "actual_otop_date" date,
    "tryout_counter" integer DEFAULT 0 NOT NULL,
    "action_plan_required" boolean DEFAULT false NOT NULL,
    "priority_level" integer DEFAULT 5,
    "notes" text,
    "created_at" timestamp with time zone DEFAULT now() NOT NULL,
    "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Core entity linking Project, Drawing, and Mold with status tracking.

Key Features:

- Unique combination per project/drawing/mold
- Automated status calculation
- OT/OTOP milestone tracking
- Tryout counter management
- Priority system (1-10 scale)

Constraints:

- Unique combination of project_id, drawing_id, mold_id
- `check_planned_dates_order`: OTOP planned date ≥ OT planned date
- `check_actual_dates_order`: OTOP actual date ≥ OT actual date
- `parent_components_priority_level_check`: Priority between 1-10
- `parent_components_tryout_counter_check`: Tryout counter ≥ 0

3.7. Action Types

sql

```
CREATE TABLE "action_types" (
  "id" bigint PRIMARY KEY,
  "action_type_name" text NOT NULL,
  "action_type_description" text,
  "triggers_tryout_counter" boolean DEFAULT false NOT NULL,
  "is_active" boolean DEFAULT true NOT NULL,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Defines standard types of actions for Action Plans.

Key Features:

- Unique action type names
- Tryout counter trigger capability
- Soft delete support

3.8. Action Plans

sql

```
CREATE TABLE "action_plans" (
  "id" bigint PRIMARY KEY,
  "parent_component_id" bigint NOT NULL,
  "action_type_id" bigint NOT NULL,
  "action_plan_description" text NOT NULL,
  "responsible_user_id" uuid,
  "due_date" date NOT NULL,
  "action_plan_status" action_plan_status_enum DEFAULT 'Open' NOT NULL,
  "priority_level" integer DEFAULT 5,
  "notes" text,
  "created_by_user_id" uuid NOT NULL,
  "completed_date" date,
  "verified_date" date,
  "verified_by_user_id" uuid,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Stores action plans to address issues for parent components.

Key Features:

- Comprehensive workflow tracking

- Responsibility assignment
- Verification process
- Priority management
- Status progression (Open → In Progress → Completed → Verified)

Constraints:

- `action_plans_priority_level_check`: Priority between 1-10
- `check_action_plan_dates_logic`: Complex date validation ensuring logical progression

3.9. Cavity Evaluations

```sql
CREATE TABLE "cavity_evaluations" (
    "id" bigint PRIMARY KEY,
    "parent_component_id" bigint NOT NULL,
    "cavity_number" integer NOT NULL,
    "evaluation_date" date NOT NULL,
    "external_report_id" text,
    "report_attachment_path" text,
    "standard_dimensions_ok_count" integer NOT NULL,
    "spc_dimensions_ok_count" integer NOT NULL,
    "cavity_notes" text,
    "is_evaluated_in_report" boolean DEFAULT true NOT NULL,
    "created_by_user_id" uuid NOT NULL,
    "created_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Records dimensional results for specific cavities at points in time.

Key Features:

- Individual cavity tracking
- Dimension compliance counting
- External report linking
- File attachment support
- Evaluation status tracking

Constraints:

- `cavity_evaluations_cavity_number_check`: Cavity number > 0 (implemented via trigger)
- `cavity_evaluations_spc_dimensions_ok_count_check`: SPC count ≥ 0
- `cavity_evaluations_standard_dimensions_ok_count_check`: Standard count ≥ 0

- check_cavity_evaluation_date: Evaluation date ≤ current date

3.10. SPC Values

```sql
CREATE TABLE "spc_values" (
    "id" bigint PRIMARY KEY,
    "cavity_evaluation_id" bigint NOT NULL,
    "spc_dimension_identifier" text NOT NULL,
    "cp_value" numeric(10,3),
    "cpk_value" numeric(10,3),
    "target_value" numeric(10,3),
    "measured_value" numeric(10,3),
    "tolerance_upper" numeric(10,3),
    "tolerance_lower" numeric(10,3)
);
```

Description: Stores Cp/Cpk and tolerance values for individual SPC dimensions.

Key Features:

- Detailed statistical process control data
- Target vs. measured value tracking
- Tolerance specification
- Capability indices (Cp/Cpk)

Constraints:

- Unique combination of cavity_evaluation_id and spc_dimension_identifier
- check_tolerance_order: Upper tolerance ≥ lower tolerance
- spc_values_cp_value_check: Cp value ≥ 0 (no Cpk constraint in current schema)

4. User Management

4.1. User Roles

```sql
```

```sql
CREATE TABLE "user_roles" (
  "id" bigint PRIMARY KEY,
  "role_name" text NOT NULL,
  "role_description" text,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Defines available roles within the system.

Key Features:

- Simple role definition without JSONB permissions
- Unique role names
- No `is_active` field

4.2. Users

sql

```sql
CREATE TABLE "users" (
  "id" uuid PRIMARY KEY,
  "full_name" text NOT NULL,
  "email" text NOT NULL,
  "is_active" boolean DEFAULT true NOT NULL,
  "last_login_at" timestamp with time zone,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

Description: Stores user profile information, extending Supabase auth.users.

Key Features:

- UUID primary key linked to Supabase auth
- Required full name and email
- Login tracking
- Soft delete via `is_active`

4.3. User Role Assignments

sql

```sql
CREATE TABLE "user_role_assignments" (
  "id" bigint PRIMARY KEY,
  "user_id" uuid NOT NULL,
  "role_id" bigint NOT NULL,
  "assigned_at" timestamp with time zone DEFAULT now() NOT NULL,
  "assigned_by_user_id" uuid
);
```

Description: Links users to their assigned roles.

Key Features:

- Multiple roles per user support
- Assignment tracking with optional assigner reference
- No `is_active` field - uses direct deletion

4.4. User Project Assignments

```sql
CREATE TABLE "user_project_assignments" (
  "id" bigint PRIMARY KEY,
  "user_id" uuid NOT NULL,
  "project_id" bigint NOT NULL,
  "assigned_at" timestamp with time zone DEFAULT now() NOT NULL,
  "assigned_by_user_id" uuid,
  "last_accessed" timestamp with time zone
);
```

Description: Links external users to projects they can access.

Key Features:

- Tracks when users last accessed projects
- Assignment tracking with optional assigner reference
- No `is_active` field - uses direct deletion

5. Milestone Management

5.1. Milestone Definitions

```sql
```

```sql
CREATE TABLE "milestone_definitions" (
  "id" bigint PRIMARY KEY,
  "milestone_name" text NOT NULL,
  "milestone_description" text,
  "milestone_order" integer NOT NULL,
  "is_active" boolean DEFAULT true NOT NULL,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

5.2. Project Milestone Instances

```sql
CREATE TABLE "project_milestone_instances" (
  "id" bigint PRIMARY KEY,
  "project_id" bigint NOT NULL,
  "milestone_definition_id" bigint NOT NULL,
  "milestone_target_date" date NOT NULL,
  "milestone_actual_date" date,
  "milestone_status" milestone_status_enum DEFAULT 'Planned' NOT NULL,
  "notes" text,
  "updated_by_user_id" uuid,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

5.3. OTOP Target Rules

```sql
CREATE TABLE "otop_target_rules" (
  "id" bigint PRIMARY KEY,
  "component_classification_id" bigint NOT NULL,
  "mold_type" mold_type_enum NOT NULL,
  "target_days_from_ot" integer NOT NULL,
  "is_active" boolean DEFAULT true NOT NULL,
  "created_at" timestamp with time zone DEFAULT now() NOT NULL,
  "updated_at" timestamp with time zone DEFAULT now() NOT NULL
);
```

6. Missing Field in Cavity Evaluations

Note: The current schema includes a `last_modified` field in `cavity_evaluations` that is referenced in triggers but not present in the main table definition. This should be added:

```sql
ALTER TABLE cavity_evaluations
ADD COLUMN last_modified timestamp with time zone DEFAULT now();
```

7. Enumerations

- `project_status_enum`: 'Active', 'Archived', 'Closed'
- `mold_type_enum`: 'Prototype', 'Low Volume', 'Definitive'
- `parent_status_enum`: 'NEW', 'INCOMPLETE_DATA', 'KO', 'NOT_OFF_TOOL', 'OTOP', 'P_OTOP', 'OT'
- `action_plan_status_enum`: 'Open', 'In Progress', 'Completed', 'Verified', 'Cancelled'
- `milestone_status_enum`: 'Planned', 'In Progress', 'Completed', 'Delayed', 'Skipped', 'Cancelled'
- `notification_type_enum`: 'ACTION_PLAN_DUE', 'STATUS_CHANGE_KO', 'NEW_ASSIGNMENT', 'MILESTONE_DELAY'
- `target_status_type_enum`: 'OT', 'OTOP'

8. Views

7.1. Action Plans Detailed View

```sql
CREATE VIEW "v_action_plans_detailed" AS
-- Comprehensive view joining action plans with related entities
```

7.2. Parent Components Dashboard View

```sql
CREATE VIEW "v_parent_components_dashboard" AS
-- Dashboard view for parent component overview
```

9. Key Relationships

- **Projects** → **Parent Components** (1:N)
- **Drawings** → **Parent Components** (1:N)
- **Molds** → **Parent Components** (1:N)
- **Parent Components** → **Cavity Evaluations** (1:N)
- **Parent Components** → **Action Plans** (1:N)
- **Cavity Evaluations** → **SPC Values** (1:N)
- **Users** ↔ **Projects** (N:M via assignments)

- **Users** ↔ **Roles** (N:M via assignments)

10. Data Integrity Functions

The schema includes a comprehensive data validation function:

```sql
CREATE FUNCTION "validate_data_integrity"()
RETURNS TABLE(validation_type text, details text)
```

This function performs system-wide data integrity checks and reports any inconsistencies.

11. Security and Permissions

- All tables grant permissions to `anon`, `authenticated`, and `service_role`
- Row Level Security (RLS) can be implemented for multi-tenant access
- User authentication handled via Supabase Auth with UUID references

12. Database Features Used

- **JSONB**: Removed from user roles - simplified permission model
- **Check Constraints**: Extensive validation at database level
- **Foreign Keys**: Referential integrity across all relationships
- **Triggers**: Automatic timestamp updates, audit logging, and status calculation
- **Views**: Pre-aggregated data for dashboard performance
- **Sequences**: Auto-incrementing primary keys
- **Comprehensive RLS**: Row-level security for multi-tenant access
- **Advanced Functions**: Complex business logic in PostgreSQL functions

This data model provides a robust foundation for the geometric maturity tracking application with comprehensive audit trails, flexible user management, and strong data integrity constraints.