-- OPTIONAL BUT RECOMMENDED EXTENSIONS

-- CREATE EXTENSION IF NOT EXISTS postgis;

-- CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;

--------------------------------------------------------------------------------

-- SCHEMA: gr33n\_core - Essential entities and logic for all gr33n instances

--------------------------------------------------------------------------------

CREATE SCHEMA IF NOT EXISTS gr33n\_core;

COMMENT ON SCHEMA gr33n\_core IS 'Core entities shared across all gr33n farm instances, including users, farms, zones, devices, sensors, actuators, tasks, schedules, automation rules, notifications, system logging, file attachments, weather data, cost tracking, validation rules, and user activity. This schema forms the backbone of the gr33n platform.';

--------------------------------------------------------------------------------

-- ENUMERATED TYPES

--------------------------------------------------------------------------------

CREATE TYPE gr33n\_core.farm\_scale\_tier\_enum AS ENUM ('small', 'medium', 'large', 'enterprise');

COMMENT ON TYPE gr33n\_core.farm\_scale\_tier\_enum IS 'Defines the scale categories for a farm, influencing UI/API behavior and default configurations.';

CREATE TYPE gr33n\_core.operational\_status\_enum AS ENUM ('active', 'maintenance', 'planning', 'archived', 'decommissioned');

COMMENT ON TYPE gr33n\_core.operational\_status\_enum IS 'Defines the current operational state of a farm.';

CREATE TYPE gr33n\_core.log\_level\_enum AS ENUM ('DEBUG', 'INFO', 'NOTICE', 'WARNING', 'ERROR', 'CRITICAL', 'ALERT', 'EMERGENCY');

COMMENT ON TYPE gr33n\_core.log\_level\_enum IS 'Defines severity levels for system log entries.';

CREATE TYPE gr33n\_core.user\_role\_enum AS ENUM ('user', 'farm\_manager', 'farm\_worker', 'gr33n\_system\_admin');

COMMENT ON TYPE gr33n\_core.user\_role\_enum IS 'Defines system-wide roles for users in the gr33n platform.';

CREATE TYPE gr33n\_core.farm\_member\_role\_enum AS ENUM ('owner', 'manager', 'agronomist', 'worker', 'viewer', 'custom\_role');

COMMENT ON TYPE gr33n\_core.farm\_member\_role\_enum IS 'Defines roles for users specifically within a farm.';

CREATE TYPE gr33n\_core.device\_status\_enum AS ENUM ('online', 'offline', 'error\_comms', 'error\_hardware', 'maintenance\_mode', 'initializing', 'unknown', 'decommissioned', 'pending\_activation');

COMMENT ON TYPE gr33n\_core.device\_status\_enum IS 'Defines the operational status of a device.';

CREATE TYPE gr33n\_core.task\_status\_enum AS ENUM ('todo', 'in\_progress', 'on\_hold', 'completed', 'cancelled', 'blocked\_requires\_input', 'pending\_review');

COMMENT ON TYPE gr33n\_core.task\_status\_enum IS 'Defines the status of a task.';

CREATE TYPE gr33n\_core.automation\_trigger\_source\_enum AS ENUM ('sensor\_reading\_threshold', 'specific\_time\_cron', 'actuator\_state\_changed', 'manual\_api\_trigger', 'task\_status\_updated', 'new\_system\_log\_event', 'external\_webhook\_received');

COMMENT ON TYPE gr33n\_core.automation\_trigger\_source\_enum IS 'Defines the types of triggers for automation rules.';

CREATE TYPE gr33n\_core.executable\_action\_type\_enum AS ENUM ('control\_actuator', 'trigger\_another\_automation\_rule', 'send\_notification', 'create\_task', 'log\_custom\_event', 'http\_webhook\_call', 'update\_record\_in\_gr33n');

COMMENT ON TYPE gr33n\_core.executable\_action\_type\_enum IS 'Defines the types of actions that can be executed by schedules or automation rules.';

CREATE TYPE gr33n\_core.notification\_priority\_enum AS ENUM ('low', 'medium', 'high', 'critical');

COMMENT ON TYPE gr33n\_core.notification\_priority\_enum IS 'Defines priority levels for notifications.';

CREATE TYPE gr33n\_core.notification\_status\_enum AS ENUM ('pending', 'queued', 'sent', 'delivered', 'failed\_to\_send', 'read\_by\_user', 'acknowledged\_by\_user', 'archived\_by\_user', 'system\_cleared');

COMMENT ON TYPE gr33n\_core.notification\_status\_enum IS 'Defines the status of a notification instance.';

CREATE TYPE gr33n\_core.actuator\_event\_source\_enum AS ENUM ('manual\_ui\_input', 'manual\_api\_call', 'schedule\_trigger', 'automation\_rule\_trigger', 'device\_internal\_feedback\_loop', 'system\_initialization\_routine', 'emergency\_stop\_signal');

COMMENT ON TYPE gr33n\_core.actuator\_event\_source\_enum IS 'Defines the source of an actuator event command.';

CREATE TYPE gr33n\_core.actuator\_execution\_status\_enum AS ENUM ('command\_sent\_to\_device', 'acknowledged\_by\_device', 'execution\_started\_on\_device', 'execution\_completed\_success\_on\_device', 'execution\_completed\_with\_error\_on\_device', 'execution\_failed\_to\_start\_on\_device', 'pending\_confirmation\_from\_feedback', 'timeout\_waiting\_for\_acknowledgement', 'cancelled\_by\_user\_or\_system');

COMMENT ON TYPE gr33n\_core.actuator\_execution\_status\_enum IS 'Defines the execution status of an actuator command.';

CREATE TYPE gr33n\_core.weather\_data\_source\_enum AS ENUM ('farm\_weather\_station', 'api\_openweather', 'api\_visualcrossing', 'manual\_entry', 'iot\_sensor\_reading');

COMMENT ON TYPE gr33n\_core.weather\_data\_source\_enum IS 'Defines the origin of the weather data.';

CREATE TYPE gr33n\_core.cost\_category\_enum AS ENUM (

'seeds\_plants', 'fertilizers\_soil\_amendments', 'pest\_disease\_control', 'water\_irrigation',

'labor\_wages', 'equipment\_purchase\_rental', 'equipment\_maintenance\_fuel', 'utilities\_electricity\_gas',

'land\_rent\_mortgage', 'insurance', 'licenses\_permits', 'feed\_livestock', 'veterinary\_services',

'packaging\_supplies', 'transportation\_logistics', 'marketing\_sales', 'training\_consultancy', 'miscellaneous'

);

COMMENT ON TYPE gr33n\_core.cost\_category\_enum IS 'Defines categories for financial cost tracking in farm operations.';

CREATE TYPE gr33n\_core.validation\_rule\_type\_enum AS ENUM ('range\_check', 'required\_field', 'format\_validation', 'regex\_match', 'lookup\_in\_list', 'cross\_field\_comparison', 'custom\_function\_check');

COMMENT ON TYPE gr33n\_core.validation\_rule\_type\_enum IS 'Defines types of data validation rules that can be applied.';

CREATE TYPE gr33n\_core.validation\_severity\_enum AS ENUM ('warning', 'error', 'critical\_stop');

COMMENT ON TYPE gr33n\_core.validation\_severity\_enum IS 'Defines the severity of a validation rule failure.';

CREATE TYPE gr33n\_core.user\_action\_type\_enum AS ENUM (

'login\_success', 'login\_failure', 'logout',

'create\_record', 'view\_record', 'update\_record', 'delete\_record', 'list\_records',

'execute\_action', 'change\_setting', 'system\_event', 'export\_data', 'import\_data'

);

COMMENT ON TYPE gr33n\_core.user\_action\_type\_enum IS 'Defines types of user actions for activity logging.';

--------------------------------------------------------------------------------

-- TABLES

--------------------------------------------------------------------------------

-- User Profiles (linked to Supabase auth.users table if used)

CREATE TABLE gr33n\_core.profiles (

user\_id UUID PRIMARY KEY REFERENCES auth.users(id) ON DELETE CASCADE,

full\_name TEXT,

email TEXT UNIQUE NOT NULL,

avatar\_url TEXT,

role gr33n\_core.user\_role\_enum DEFAULT 'user' NOT NULL,

preferences JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

-- Note: 'updated\_by\_user\_id' usually not needed here as profile updates are self-initiated or by admin (logged in activity log)

-- Note: 'deleted\_at' for users is complex due to auth.users; usually means deactivation/anonymization.

);

COMMENT ON TABLE gr33n\_core.profiles IS 'Stores user-specific information, application preferences, and extends Supabase''s auth.users table.';

-- Farms

CREATE TABLE gr33n\_core.farms (

id BIGSERIAL PRIMARY KEY,

name TEXT NOT NULL,

description TEXT,

location\_text TEXT,

-- location\_gis GEOMETRY(Point, 4326),

size\_hectares NUMERIC(10,2),

farm\_type TEXT,

scale\_tier gr33n\_core.farm\_scale\_tier\_enum DEFAULT 'small' NOT NULL,

owner\_user\_id UUID NOT NULL REFERENCES gr33n\_core.profiles(user\_id),

timezone TEXT DEFAULT 'UTC' NOT NULL,

currency CHAR(3) DEFAULT 'USD' NOT NULL,

operational\_status gr33n\_core.operational\_status\_enum DEFAULT 'active' NOT NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL, -- ADDED

deleted\_at TIMESTAMPTZ DEFAULT NULL -- ADDED for soft delete

);

COMMENT ON TABLE gr33n\_core.farms IS 'Represents a distinct agricultural operation. Includes soft delete and last updater.';

-- Farm Memberships

CREATE TABLE gr33n\_core.farm\_memberships (

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

user\_id UUID NOT NULL REFERENCES gr33n\_core.profiles(user\_id) ON DELETE CASCADE,

role\_in\_farm gr33n\_core.farm\_member\_role\_enum NOT NULL,

permissions JSONB DEFAULT '{}'::jsonb,

joined\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

-- No updated\_at/by here as changes are essentially new states or deletions.

-- No deleted\_at here; removing a membership is a hard delete of the relationship.

PRIMARY KEY (farm\_id, user\_id)

);

COMMENT ON TABLE gr33n\_core.farm\_memberships IS 'Manages user access rights and their specific roles within a particular farm.';

-- Farm Active Modules

CREATE TABLE gr33n\_core.farm\_active\_modules (

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

module\_schema\_name TEXT NOT NULL,

is\_enabled BOOLEAN DEFAULT TRUE NOT NULL,

configuration JSONB DEFAULT '{}'::jsonb,

activated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

-- updated\_at/by managed by application logic if module config changes

PRIMARY KEY (farm\_id, module\_schema\_name)

);

COMMENT ON TABLE gr33n\_core.farm\_active\_modules IS 'Defines which gr33n modules are enabled and configured for a specific farm.';

-- Zones

CREATE TABLE gr33n\_core.zones (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

parent\_zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL,

name TEXT NOT NULL,

description TEXT,

zone\_type TEXT,

area\_sqm NUMERIC(12,2),

-- boundary\_gis GEOMETRY(Polygon, 4326),

meta\_data JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL, -- ADDED

deleted\_at TIMESTAMPTZ DEFAULT NULL -- ADDED for soft delete

);

COMMENT ON TABLE gr33n\_core.zones IS 'Defines distinct areas within a farm. Includes soft delete and last updater.';

-- Devices

CREATE TABLE gr33n\_core.devices (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Zone can be soft-deleted

name TEXT NOT NULL,

device\_uid TEXT UNIQUE,

device\_type TEXT NOT NULL,

ip\_address INET,

firmware\_version TEXT,

status gr33n\_core.device\_status\_enum DEFAULT 'unknown' NOT NULL,

last\_heartbeat TIMESTAMPTZ,

api\_key TEXT UNIQUE,

config JSONB DEFAULT '{}'::jsonb,

meta\_data JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL, -- ADDED

deleted\_at TIMESTAMPTZ DEFAULT NULL -- ADDED for soft delete

);

COMMENT ON TABLE gr33n\_core.devices IS 'Physical hardware controllers. Includes soft delete and last updater.';

-- Sensors

CREATE TABLE gr33n\_core.sensors (

id BIGSERIAL PRIMARY KEY,

device\_id BIGINT REFERENCES gr33n\_core.devices(id) ON DELETE SET NULL, -- Device can be soft-deleted

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Zone can be soft-deleted

name TEXT NOT NULL,

sensor\_type TEXT NOT NULL,

unit\_of\_measurement TEXT,

hardware\_identifier TEXT,

value\_min\_expected NUMERIC,

value\_max\_expected NUMERIC,

alert\_threshold\_low NUMERIC,

alert\_threshold\_high NUMERIC,

reading\_interval\_seconds INTEGER,

is\_calibrated BOOLEAN DEFAULT FALSE,

last\_calibration\_date DATE,

calibration\_data JSONB,

config JSONB DEFAULT '{}'::jsonb,

meta\_data JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL, -- ADDED

deleted\_at TIMESTAMPTZ DEFAULT NULL, -- ADDED for soft delete

CONSTRAINT chk\_sensor\_farm\_context CHECK (device\_id IS NOT NULL OR farm\_id IS NOT NULL)

);

COMMENT ON TABLE gr33n\_core.sensors IS 'Individual sensors collecting data. Includes soft delete and last updater.';

-- Sensor Readings (Time-series, less likely to be soft-deleted individually, usually archived/purged by policy)

CREATE TABLE gr33n\_core.sensor\_readings (

sensor\_id BIGINT NOT NULL REFERENCES gr33n\_core.sensors(id) ON DELETE CASCADE, -- If sensor is hard deleted, readings go. If soft, they remain.

reading\_time TIMESTAMPTZ NOT NULL,

value\_numeric NUMERIC,

value\_text TEXT,

value\_json JSONB,

battery\_level\_percent NUMERIC(5,2) CHECK (battery\_level\_percent IS NULL OR (battery\_level\_percent >= 0 AND battery\_level\_percent <= 100)),

signal\_strength\_dbm INTEGER,

is\_valid BOOLEAN DEFAULT TRUE,

meta\_data JSONB DEFAULT '{}'::jsonb,

PRIMARY KEY (sensor\_id, reading\_time)

);

COMMENT ON TABLE gr33n\_core.sensor\_readings IS 'Stores time-series data from sensors. Recommended for TimescaleDB hypertable conversion.';

-- Actuators

CREATE TABLE gr33n\_core.actuators (

id BIGSERIAL PRIMARY KEY,

device\_id BIGINT NOT NULL REFERENCES gr33n\_core.devices(id) ON DELETE SET NULL, -- Device can be soft-deleted

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Zone can be soft-deleted

name TEXT NOT NULL,

actuator\_type TEXT NOT NULL,

hardware\_identifier TEXT,

current\_state\_numeric NUMERIC,

current\_state\_text TEXT,

last\_known\_state\_time TIMESTAMPTZ,

last\_command\_sent\_time TIMESTAMPTZ,

feedback\_sensor\_id BIGINT REFERENCES gr33n\_core.sensors(id) ON DELETE SET NULL, -- Sensor can be soft-deleted

config JSONB DEFAULT '{}'::jsonb,

meta\_data JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL, -- ADDED

deleted\_at TIMESTAMPTZ DEFAULT NULL -- ADDED for soft delete

);

COMMENT ON TABLE gr33n\_core.actuators IS 'Represents physical devices that perform controllable actions. Includes soft delete and last updater.';

-- Actuator Events (Time-series, less likely to be soft-deleted individually)

CREATE TABLE gr33n\_core.actuator\_events (

actuator\_id BIGINT NOT NULL REFERENCES gr33n\_core.actuators(id) ON DELETE CASCADE, -- If actuator is hard deleted, events go.

event\_time TIMESTAMPTZ NOT NULL,

command\_sent TEXT,

parameters\_sent JSONB,

triggered\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

triggered\_by\_schedule\_id BIGINT, -- FK added later

triggered\_by\_rule\_id BIGINT, -- FK added later

source gr33n\_core.actuator\_event\_source\_enum NOT NULL,

response\_received\_from\_device TEXT,

execution\_status gr33n\_core.actuator\_execution\_status\_enum,

resulting\_state\_numeric\_actual NUMERIC,

resulting\_state\_text\_actual TEXT,

meta\_data JSONB DEFAULT '{}'::jsonb,

PRIMARY KEY (actuator\_id, event\_time)

);

COMMENT ON TABLE gr33n\_core.actuator\_events IS 'Logs commands issued to actuators and their outcomes. Recommended for TimescaleDB.';

-- Tasks

CREATE TABLE gr33n\_core.tasks (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Zone can be soft-deleted

title TEXT NOT NULL,

description TEXT,

task\_type TEXT,

status gr33n\_core.task\_status\_enum DEFAULT 'todo' NOT NULL,

priority INTEGER DEFAULT 1 CHECK (priority BETWEEN 0 AND 3),

assigned\_to\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

due\_date DATE,

estimated\_duration\_minutes INTEGER,

actual\_start\_time TIMESTAMPTZ,

actual\_end\_time TIMESTAMPTZ,

related\_module\_schema TEXT,

related\_table\_name TEXT,

related\_record\_id BIGINT, -- Assuming related records (e.g. plant\_batch) might have BIGINT PKs. If mixed, this also might need to be TEXT.

created\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL, -- ADDED

deleted\_at TIMESTAMPTZ DEFAULT NULL -- ADDED for soft delete

);

COMMENT ON TABLE gr33n\_core.tasks IS 'Manages farm tasks. Includes soft delete, creator, and last updater.';

-- Schedules

CREATE TABLE gr33n\_core.schedules (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

name TEXT NOT NULL,

description TEXT,

schedule\_type TEXT NOT NULL,

cron\_expression TEXT NOT NULL,

timezone TEXT DEFAULT 'UTC' NOT NULL,

is\_active BOOLEAN DEFAULT TRUE NOT NULL,

last\_triggered\_time TIMESTAMPTZ,

next\_expected\_trigger\_time TIMESTAMPTZ,

meta\_data JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

-- updated\_by\_user\_id could be added

-- deleted\_at could be added (for deactivating vs removing schedule history)

);

COMMENT ON TABLE gr33n\_core.schedules IS 'Defines recurring schedules for automated farm actions or system task generation.';

-- Automation Rules

CREATE TABLE gr33n\_core.automation\_rules (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

name TEXT NOT NULL,

description TEXT,

is\_active BOOLEAN DEFAULT TRUE NOT NULL,

trigger\_source gr33n\_core.automation\_trigger\_source\_enum NOT NULL,

trigger\_configuration JSONB NOT NULL,

condition\_logic TEXT DEFAULT 'ALL' CHECK (condition\_logic IN ('ALL', 'ANY')),

conditions\_jsonb JSONB DEFAULT '[]'::jsonb,

last\_evaluated\_time TIMESTAMPTZ,

last\_triggered\_time TIMESTAMPTZ,

cooldown\_period\_seconds INTEGER DEFAULT 0,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

-- updated\_by\_user\_id could be added

-- deleted\_at could be added

);

COMMENT ON TABLE gr33n\_core.automation\_rules IS 'Defines complex "if-this-then-that" (IFTTT) automation logic.';

-- Executable Actions

CREATE TABLE gr33n\_core.executable\_actions (

id BIGSERIAL PRIMARY KEY,

schedule\_id BIGINT REFERENCES gr33n\_core.schedules(id) ON DELETE CASCADE,

rule\_id BIGINT REFERENCES gr33n\_core.automation\_rules(id) ON DELETE CASCADE,

execution\_order INTEGER DEFAULT 0 NOT NULL,

action\_type gr33n\_core.executable\_action\_type\_enum NOT NULL,

target\_actuator\_id BIGINT REFERENCES gr33n\_core.actuators(id) ON DELETE SET NULL, -- Actuator can be soft-deleted

target\_automation\_rule\_id BIGINT REFERENCES gr33n\_core.automation\_rules(id) ON DELETE CASCADE, -- Chained rule

target\_notification\_template\_id BIGINT, -- FK to notification\_templates, added later

action\_command TEXT,

action\_parameters JSONB,

delay\_before\_execution\_seconds INTEGER DEFAULT 0,

CONSTRAINT chk\_executable\_source CHECK (schedule\_id IS NOT NULL OR rule\_id IS NOT NULL),

CONSTRAINT chk\_executable\_action\_details CHECK (

(action\_type = 'control\_actuator' AND target\_actuator\_id IS NOT NULL AND action\_command IS NOT NULL) OR

(action\_type = 'trigger\_another\_automation\_rule' AND target\_automation\_rule\_id IS NOT NULL) OR

(action\_type = 'send\_notification' AND target\_notification\_template\_id IS NOT NULL) OR

(action\_type = 'create\_task' AND action\_parameters IS NOT NULL) OR

(action\_type = 'log\_custom\_event' AND action\_parameters IS NOT NULL) OR

(action\_type = 'http\_webhook\_call' AND action\_parameters->>'url' IS NOT NULL) OR

(action\_type = 'update\_record\_in\_gr33n' AND action\_parameters->>'target\_module\_schema' IS NOT NULL AND action\_parameters->>'target\_table\_name' IS NOT NULL AND action\_parameters->'fields\_to\_update' IS NOT NULL)

)

);

COMMENT ON TABLE gr33n\_core.executable\_actions IS 'Specifies actions for schedules or automation rules.';

-- Add FKs from actuator\_events to schedules and automation\_rules (now that these tables are defined)

ALTER TABLE gr33n\_core.actuator\_events ADD CONSTRAINT fk\_actuator\_event\_schedule FOREIGN KEY (triggered\_by\_schedule\_id) REFERENCES gr33n\_core.schedules(id) ON DELETE SET NULL;

ALTER TABLE gr33n\_core.actuator\_events ADD CONSTRAINT fk\_actuator\_event\_rule FOREIGN KEY (triggered\_by\_rule\_id) REFERENCES gr33n\_core.automation\_rules(id) ON DELETE SET NULL;

-- Notification Templates

CREATE TABLE gr33n\_core.notification\_templates (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

template\_key TEXT NOT NULL,

description TEXT,

subject\_template TEXT,

body\_template\_text TEXT,

body\_template\_html TEXT,

default\_delivery\_channels TEXT[] DEFAULT ARRAY['in\_app', 'email']::TEXT[],

default\_priority gr33n\_core.notification\_priority\_enum DEFAULT 'medium',

is\_system\_template BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

-- updated\_by\_user\_id could be added for farm-specific templates

-- deleted\_at could be added for farm-specific templates

CONSTRAINT uq\_notification\_template\_key UNIQUE (farm\_id, template\_key),

CONSTRAINT uq\_system\_notification\_template\_key UNIQUE (template\_key) WHERE farm\_id IS NULL

);

COMMENT ON TABLE gr33n\_core.notification\_templates IS 'Reusable templates for generating standardized notifications and alerts.';

ALTER TABLE gr33n\_core.executable\_actions ADD CONSTRAINT fk\_action\_notification\_template FOREIGN KEY (target\_notification\_template\_id) REFERENCES gr33n\_core.notification\_templates(id) ON DELETE SET NULL;

-- Alerts & Notifications (Log of sent/pending notifications)

CREATE TABLE gr33n\_core.alerts\_notifications (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

recipient\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

notification\_template\_id BIGINT REFERENCES gr33n\_core.notification\_templates(id) ON DELETE SET NULL,

triggering\_event\_source\_type TEXT,

triggering\_event\_source\_id BIGINT,

severity gr33n\_core.notification\_priority\_enum DEFAULT 'medium',

subject\_rendered TEXT,

message\_text\_rendered TEXT,

message\_html\_rendered TEXT,

delivery\_attempts JSONB DEFAULT '{}'::jsonb,

status gr33n\_core.notification\_status\_enum DEFAULT 'pending',

is\_read BOOLEAN DEFAULT FALSE,

read\_at TIMESTAMPTZ,

is\_acknowledged BOOLEAN DEFAULT FALSE,

acknowledged\_at TIMESTAMPTZ,

acknowledged\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

scheduled\_send\_at TIMESTAMPTZ DEFAULT NOW()

);

COMMENT ON TABLE gr33n\_core.alerts\_notifications IS 'Records generated alerts and notifications, their delivery status, and user interactions.';

-- System Logs

CREATE TABLE gr33n\_core.system\_logs (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT REFERENCES gr33n\_core.farms(id) ON DELETE SET NULL, -- Farm can be soft-deleted

user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

log\_time TIMESTAMPTZ DEFAULT NOW() NOT NULL,

log\_level gr33n\_core.log\_level\_enum NOT NULL,

event\_type TEXT,

message TEXT NOT NULL,

source\_component TEXT,

context\_data JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL -- For table metadata, log\_time is the event time

);

COMMENT ON TABLE gr33n\_core.system\_logs IS 'Stores application-level logs for system events, errors, audit trails, and operational monitoring.';

-- File Attachments

CREATE TABLE gr33n\_core.file\_attachments (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

related\_module\_schema TEXT NOT NULL,

related\_table\_name TEXT NOT NULL,

related\_record\_id TEXT NOT NULL, -- MODIFIED to TEXT for flexibility (can store BIGINT or UUID as string)

file\_name TEXT NOT NULL,

file\_type TEXT NOT NULL,

file\_size\_bytes BIGINT,

storage\_path TEXT NOT NULL,

mime\_type TEXT,

description TEXT,

uploaded\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

-- deleted\_at could be added for soft-deleting attachment records (file itself might remain in storage or be cleaned up by a process)

);

COMMENT ON TABLE gr33n\_core.file\_attachments IS 'Manages file uploads. related\_record\_id is TEXT for flexibility with different PK types.';

-- Weather Data

CREATE TABLE gr33n\_core.weather\_data (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Zone can be soft-deleted

recorded\_at TIMESTAMPTZ NOT NULL,

data\_source gr33n\_core.weather\_data\_source\_enum NOT NULL,

source\_sensor\_id BIGINT REFERENCES gr33n\_core.sensors(id) ON DELETE SET NULL, -- Sensor can be soft-deleted

temperature\_celsius NUMERIC(5,2),

humidity\_percent NUMERIC(5,2) CHECK (humidity\_percent IS NULL OR (humidity\_percent >= 0 AND humidity\_percent <= 100)),

precipitation\_mm NUMERIC(6,2),

wind\_speed\_ms NUMERIC(5,2),

wind\_direction\_degrees INTEGER CHECK (wind\_direction\_degrees IS NULL OR (wind\_direction\_degrees >= 0 AND wind\_direction\_degrees <= 360)),

barometric\_pressure\_hpa NUMERIC(7,2),

solar\_radiation\_wm2 NUMERIC(8,2),

dew\_point\_celsius NUMERIC(5,2),

uv\_index NUMERIC(4,1) CHECK (uv\_index IS NULL OR (uv\_index >=0)),

cloud\_cover\_percent NUMERIC(5,2) CHECK (cloud\_cover\_percent IS NULL OR (cloud\_cover\_percent >= 0 AND cloud\_cover\_percent <= 100)),

forecast\_data JSONB,

raw\_data JSONB,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

CONSTRAINT uq\_weather\_data\_key UNIQUE (farm\_id, zone\_id, recorded\_at, data\_source, source\_sensor\_id)

);

COMMENT ON TABLE gr33n\_core.weather\_data IS 'Stores historical and potentially forecasted weather data. Recommended for TimescaleDB.';

-- Cost Transactions

CREATE TABLE gr33n\_core.cost\_transactions (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

transaction\_date DATE NOT NULL,

category gr33n\_core.cost\_category\_enum NOT NULL,

subcategory TEXT,

amount NUMERIC(12,2) NOT NULL,

currency CHAR(3) NOT NULL, -- App ensures this aligns with farm's currency preference

description TEXT,

related\_module\_schema TEXT,

related\_table\_name TEXT,

related\_record\_id BIGINT, -- Assuming costs usually link to entities with BIGINT PKs. If mixed, could be TEXT.

receipt\_file\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL,

is\_income BOOLEAN DEFAULT FALSE NOT NULL,

created\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id),

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

-- updated\_by\_user\_id could be added

-- deleted\_at could be added

);

COMMENT ON TABLE gr33n\_core.cost\_transactions IS 'Tracks financial costs and income. Currency should align with farm settings via application logic.';

-- Validation Rules

CREATE TABLE gr33n\_core.validation\_rules (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

rule\_name TEXT NOT NULL, -- Should be unique per farm or globally if farm\_id is NULL

description TEXT,

target\_module\_schema TEXT NOT NULL,

target\_table\_name TEXT NOT NULL,

target\_column\_name TEXT NOT NULL,

rule\_type gr33n\_core.validation\_rule\_type\_enum NOT NULL,

rule\_config JSONB NOT NULL,

error\_message\_template TEXT,

is\_active BOOLEAN DEFAULT TRUE NOT NULL,

severity gr33n\_core.validation\_severity\_enum DEFAULT 'error' NOT NULL,

evaluation\_trigger TEXT DEFAULT 'on\_save' NOT NULL CHECK (evaluation\_trigger IN ('on\_save', 'on\_change', 'manual\_batch')),

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

-- updated\_by\_user\_id could be added

-- deleted\_at could be added

CONSTRAINT uq\_validation\_rule\_name\_farm UNIQUE (farm\_id, rule\_name),

CONSTRAINT uq\_validation\_rule\_name\_global UNIQUE (rule\_name) WHERE farm\_id IS NULL

);

COMMENT ON TABLE gr33n\_core.validation\_rules IS 'Defines data validation rules. rule\_name is unique per farm or globally.';

-- User Activity Log

CREATE TABLE gr33n\_core.user\_activity\_log (

id BIGSERIAL PRIMARY KEY,

user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

farm\_id BIGINT REFERENCES gr33n\_core.farms(id) ON DELETE SET NULL, -- Farm can be soft-deleted

activity\_time TIMESTAMPTZ DEFAULT NOW() NOT NULL,

action\_type gr33n\_core.user\_action\_type\_enum NOT NULL,

target\_module\_schema TEXT,

target\_table\_name TEXT,

target\_record\_id TEXT, -- Flexible for various PK types

target\_record\_description TEXT,

ip\_address INET,

user\_agent TEXT,

session\_id TEXT,

status TEXT CHECK (status IN ('success', 'failure', 'pending')),

failure\_reason TEXT,

details JSONB DEFAULT '{}'::jsonb,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

);

COMMENT ON TABLE gr33n\_core.user\_activity\_log IS 'Logs user actions and significant system events. Recommended for TimescaleDB.';

--------------------------------------------------------------------------------

-- INDEXING STRATEGY

--------------------------------------------------------------------------------

-- Existing Indexes (Primary Keys are automatically indexed)

-- Indexes for soft deletes (example for farms, apply similarly to other tables with deleted\_at)

CREATE INDEX IF NOT EXISTS idx\_farms\_deleted\_at ON gr33n\_core.farms(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

CREATE INDEX IF NOT EXISTS idx\_zones\_deleted\_at ON gr33n\_core.zones(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

CREATE INDEX IF NOT EXISTS idx\_devices\_deleted\_at ON gr33n\_core.devices(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

CREATE INDEX IF NOT EXISTS idx\_sensors\_deleted\_at ON gr33n\_core.sensors(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

CREATE INDEX IF NOT EXISTS idx\_actuators\_deleted\_at ON gr33n\_core.actuators(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

CREATE INDEX IF NOT EXISTS idx\_tasks\_deleted\_at ON gr33n\_core.tasks(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- Indexes for Sensor Readings

CREATE INDEX IF NOT EXISTS idx\_sensor\_readings\_time\_desc ON gr33n\_core.sensor\_readings(reading\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_sensor\_readings\_value\_numeric ON gr33n\_core.sensor\_readings(value\_numeric) WHERE value\_numeric IS NOT NULL;

-- Indexes for Actuator Events

CREATE INDEX IF NOT EXISTS idx\_actuator\_events\_time\_desc ON gr33n\_core.actuator\_events(event\_time DESC);

-- Indexes for Farms

CREATE INDEX IF NOT EXISTS idx\_farms\_owner\_lookup ON gr33n\_core.farms(owner\_user\_id);

CREATE INDEX IF NOT EXISTS idx\_farms\_name ON gr33n\_core.farms(name text\_pattern\_ops);

CREATE INDEX IF NOT EXISTS idx\_farms\_updated\_by ON gr33n\_core.farms(updated\_by\_user\_id);

-- Indexes for Zones

CREATE INDEX IF NOT EXISTS idx\_zones\_farm\_id ON gr33n\_core.zones(farm\_id);

CREATE INDEX IF NOT EXISTS idx\_zones\_parent\_id ON gr33n\_core.zones(parent\_zone\_id);

CREATE INDEX IF NOT EXISTS idx\_zones\_updated\_by ON gr33n\_core.zones(updated\_by\_user\_id);

-- Indexes for Devices

CREATE INDEX IF NOT EXISTS idx\_devices\_farm\_zone ON gr33n\_core.devices(farm\_id, zone\_id);

CREATE INDEX IF NOT EXISTS idx\_devices\_status ON gr33n\_core.devices(farm\_id, status);

CREATE INDEX IF NOT EXISTS idx\_devices\_type ON gr33n\_core.devices(farm\_id, device\_type);

CREATE INDEX IF NOT EXISTS idx\_devices\_updated\_by ON gr33n\_core.devices(updated\_by\_user\_id);

-- Indexes for Sensors

CREATE INDEX IF NOT EXISTS idx\_sensors\_farm\_type ON gr33n\_core.sensors(farm\_id, sensor\_type);

CREATE INDEX IF NOT EXISTS idx\_sensors\_farm\_zone ON gr33n\_core.sensors(farm\_id, zone\_id);

CREATE INDEX IF NOT EXISTS idx\_sensors\_device\_id ON gr33n\_core.sensors(device\_id);

CREATE INDEX IF NOT EXISTS idx\_sensors\_updated\_by ON gr33n\_core.sensors(updated\_by\_user\_id);

-- Indexes for Actuators

CREATE INDEX IF NOT EXISTS idx\_actuators\_farm\_id ON gr33n\_core.actuators(farm\_id);

CREATE INDEX IF NOT EXISTS idx\_actuators\_device\_id ON gr33n\_core.actuators(device\_id);

CREATE INDEX IF NOT EXISTS idx\_actuators\_updated\_by ON gr33n\_core.actuators(updated\_by\_user\_id);

-- Indexes for Tasks

CREATE INDEX IF NOT EXISTS idx\_tasks\_assignment ON gr33n\_core.tasks(farm\_id, assigned\_to\_user\_id, status);

CREATE INDEX IF NOT EXISTS idx\_tasks\_status\_due\_date ON gr33n\_core.tasks(farm\_id, status, due\_date);

CREATE INDEX IF NOT EXISTS idx\_tasks\_related\_entity ON gr33n\_core.tasks(related\_module\_schema, related\_table\_name, related\_record\_id);

CREATE INDEX IF NOT EXISTS idx\_tasks\_created\_by ON gr33n\_core.tasks(created\_by\_user\_id);

CREATE INDEX IF NOT EXISTS idx\_tasks\_updated\_by ON gr33n\_core.tasks(updated\_by\_user\_id);

-- Indexes for Alerts & Notifications

CREATE INDEX IF NOT EXISTS idx\_alerts\_user\_status ON gr33n\_core.alerts\_notifications(recipient\_user\_id, status, created\_at DESC);

CREATE INDEX IF NOT EXISTS idx\_alerts\_farm\_status ON gr33n\_core.alerts\_notifications(farm\_id, status, created\_at DESC);

CREATE INDEX IF NOT EXISTS idx\_alerts\_template\_id ON gr33n\_core.alerts\_notifications(notification\_template\_id);

-- Indexes for System Logs

CREATE INDEX IF NOT EXISTS idx\_system\_logs\_farm\_time ON gr33n\_core.system\_logs(farm\_id, log\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_system\_logs\_level\_time ON gr33n\_core.system\_logs(log\_level, log\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_system\_logs\_event\_type\_time ON gr33n\_core.system\_logs(event\_type, log\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_system\_logs\_user\_id ON gr33n\_core.system\_logs(user\_id);

-- Indexes for File Attachments

CREATE INDEX IF NOT EXISTS idx\_file\_attachments\_related\_entity ON gr33n\_core.file\_attachments(related\_module\_schema, related\_table\_name, related\_record\_id);

CREATE INDEX IF NOT EXISTS idx\_file\_attachments\_farm\_id ON gr33n\_core.file\_attachments(farm\_id);

CREATE INDEX IF NOT EXISTS idx\_file\_attachments\_uploader ON gr33n\_core.file\_attachments(uploaded\_by\_user\_id);

-- Indexes for Weather Data

CREATE INDEX IF NOT EXISTS idx\_weather\_data\_farm\_zone\_time ON gr33n\_core.weather\_data(farm\_id, zone\_id, recorded\_at DESC);

CREATE INDEX IF NOT EXISTS idx\_weather\_data\_recorded\_at ON gr33n\_core.weather\_data(recorded\_at DESC);

-- Indexes for Cost Transactions

CREATE INDEX IF NOT EXISTS idx\_cost\_transactions\_farm\_date ON gr33n\_core.cost\_transactions(farm\_id, transaction\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_cost\_transactions\_category ON gr33n\_core.cost\_transactions(farm\_id, category, subcategory);

CREATE INDEX IF NOT EXISTS idx\_cost\_transactions\_related\_entity ON gr33n\_core.cost\_transactions(related\_module\_schema, related\_table\_name, related\_record\_id);

CREATE INDEX IF NOT EXISTS idx\_cost\_transactions\_receipt ON gr33n\_core.cost\_transactions(receipt\_file\_id);

CREATE INDEX IF NOT EXISTS idx\_cost\_transactions\_created\_by ON gr33n\_core.cost\_transactions(created\_by\_user\_id);

-- Indexes for Validation Rules

CREATE INDEX IF NOT EXISTS idx\_validation\_rules\_target ON gr33n\_core.validation\_rules(target\_module\_schema, target\_table\_name, target\_column\_name);

CREATE INDEX IF NOT EXISTS idx\_validation\_rules\_farm\_id ON gr33n\_core.validation\_rules(farm\_id);

-- Indexes for User Activity Log

CREATE INDEX IF NOT EXISTS idx\_user\_activity\_log\_user\_time ON gr33n\_core.user\_activity\_log(user\_id, activity\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_user\_activity\_log\_farm\_time ON gr33n\_core.user\_activity\_log(farm\_id, activity\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_user\_activity\_log\_action\_type ON gr33n\_core.user\_activity\_log(action\_type, activity\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_user\_activity\_log\_target ON gr33n\_core.user\_activity\_log(target\_module\_schema, target\_table\_name, target\_record\_id);

--------------------------------------------------------------------------------

-- TIMESCALEDB HYPERTABLE CONVERSIONS (Example commands, run after extension enabled and tables created)

--------------------------------------------------------------------------------

-- SELECT create\_hypertable('gr33n\_core.sensor\_readings', 'reading\_time', if\_not\_exists => TRUE, chunk\_time\_interval => INTERVAL '1 day');

-- SELECT create\_hypertable('gr33n\_core.actuator\_events', 'event\_time', if\_not\_exists => TRUE, chunk\_time\_interval => INTERVAL '1 day');

-- SELECT create\_hypertable('gr33n\_core.weather\_data', 'recorded\_at', if\_not\_exists => TRUE, chunk\_time\_interval => INTERVAL '7 days');

-- SELECT create\_hypertable('gr33n\_core.user\_activity\_log', 'activity\_time', if\_not\_exists => TRUE, chunk\_time\_interval => INTERVAL '7 days');

-- SELECT create\_hypertable('gr33n\_core.system\_logs', 'log\_time', if\_not\_exists => TRUE, chunk\_time\_interval => INTERVAL '7 days');

--------------------------------------------------------------------------------

-- SCHEMA: gr33n\_natural\_farming - Natural farming inputs, recipes, and batches

--------------------------------------------------------------------------------

CREATE SCHEMA IF NOT EXISTS gr33n\_natural\_farming;

COMMENT ON SCHEMA gr33n\_natural\_farming IS 'Manages definitions, recipes, batches, and usage guidelines for natural farming inputs (e.g., JADAM, KNF style ferments, microbial cultures).';

--------------------------------------------------------------------------------

-- ENUMERATED TYPES for gr33n\_natural\_farming

--------------------------------------------------------------------------------

CREATE TYPE gr33n\_natural\_farming.input\_category\_enum AS ENUM (

'microbial\_inoculant', -- e.g., IMO (Indigenous Microorganisms), LAB (Lactic Acid Bacteria)

'fermented\_plant\_juice', -- e.g., FPJ, JADAM JLF (Leaf Mold Fertilizer)

'water\_soluble\_nutrient',-- e.g., WCA (Water Soluble Calcium), WCP (Water Soluble Phosphorus)

'oriental\_herbal\_nutrient',-- e.g., OHN

'fish\_amino\_acid', -- e.g., FAA

'insect\_attractant\_repellent', -- e.g., JADAM JS (Jadam Sulfur), JADAM JWA (Jadam Wetting Agent)

'soil\_conditioner',

'compost\_tea\_extract',

'biochar\_preparation',

'other\_ferment',

'other\_extract'

);

COMMENT ON TYPE gr33n\_natural\_farming.input\_category\_enum IS 'Categorization of natural farming inputs.';

CREATE TYPE gr33n\_natural\_farming.input\_batch\_status\_enum AS ENUM (

'planning', 'ingredients\_gathered', 'mixing\_in\_progress', 'fermenting\_brewing',

'maturing\_aging', 'ready\_for\_use', 'partially\_used', 'fully\_used', 'expired\_discarded', 'failed\_production'

);

COMMENT ON TYPE gr33n\_natural\_farming.input\_batch\_status\_enum IS 'Lifecycle status of a specific batch of natural farming input.';

CREATE TYPE gr33n\_natural\_farming.application\_target\_enum AS ENUM (

'soil\_drench', 'foliar\_spray', 'seed\_treatment', 'compost\_pile\_inoculant', 'livestock\_water\_supplement', 'other'

);

COMMENT ON TYPE gr33n\_natural\_farming.application\_target\_enum IS 'Intended application method/target for an input or recipe.';

--------------------------------------------------------------------------------

-- TABLES for gr33n\_natural\_farming

--------------------------------------------------------------------------------

-- Input Definitions: Master list of natural farming inputs the farm uses or knows about.

CREATE TABLE gr33n\_natural\_farming.input\_definitions (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

name TEXT NOT NULL, -- e.g., "JADAM Leaf Mold Fertilizer (JLF)", "Indigenous Microorganisms Level 3 (IMO3)"

category gr33n\_natural\_farming.input\_category\_enum NOT NULL,

description TEXT, -- Purpose, benefits, general characteristics

typical\_ingredients TEXT, -- General list of ingredients often used

preparation\_summary TEXT, -- Brief overview of the making process

storage\_guidelines TEXT,

safety\_precautions TEXT,

reference\_source TEXT, -- e.g., "JADAM Organic Farming book page X", "KNF Manual Chapter Y"

file\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL, -- For detailed recipe documents or photos

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_input\_definition\_farm\_name UNIQUE (farm\_id, name, deleted\_at)

);

COMMENT ON TABLE gr33n\_natural\_farming.input\_definitions IS 'Catalog of natural farming input types used or referenced by the farm, with general information.';

-- Input Batches: Specific batches of natural farming inputs that have been made or are in progress.

CREATE TABLE gr33n\_natural\_farming.input\_batches (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

input\_definition\_id BIGINT NOT NULL REFERENCES gr33n\_natural\_farming.input\_definitions(id) ON DELETE RESTRICT,

batch\_identifier TEXT, -- User-defined or system-generated ID for this specific batch, e.g., "JLF-Lettuce-20240528"

creation\_start\_date DATE NOT NULL,

creation\_end\_date DATE, -- Date fermentation/preparation is considered complete

expected\_ready\_date DATE,

actual\_ready\_date DATE,

quantity\_produced NUMERIC(10,2),

quantity\_unit TEXT, -- e.g., "liters", "gallons", "kg", "grams"

current\_quantity\_remaining NUMERIC(10,2), -- For tracking usage

status gr33n\_natural\_farming.input\_batch\_status\_enum DEFAULT 'planning' NOT NULL,

storage\_location TEXT,

shelf\_life\_days INTEGER, -- Estimated shelf life after being ready

ph\_value NUMERIC(4,2), -- If measured

ec\_value\_ms\_cm NUMERIC(6,2), -- Electrical conductivity if measured

temperature\_during\_making TEXT, -- e.g., "Ambient room temp", "Controlled 25C"

ingredients\_used TEXT, -- Specific ingredients and quantities for THIS batch (can be free text or structured JSONB)

-- OR, for more structured ingredients:

-- ingredients\_breakdown JSONB DEFAULT '{}'::jsonb, -- e.g., [{"ingredient\_name": "Leaf Mold", "quantity": "5kg"}, {"item\_id\_from\_inventory": 123, "quantity\_used": "2kg"}]

procedure\_followed TEXT, -- Notes on the specific steps taken for this batch

observations\_notes TEXT, -- Any observations during making (smell, color, activity)

made\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL, -- Task for making this batch

file\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL, -- Photos of the batch

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_input\_batch\_farm\_identifier UNIQUE (farm\_id, batch\_identifier, deleted\_at)

);

COMMENT ON TABLE gr33n\_natural\_farming.input\_batches IS 'Tracks specific instances (batches) of natural farming inputs being made, their status, and properties.';

-- Dilution/Application Recipes: How to dilute/mix input batches for application.

CREATE TABLE gr33n\_natural\_farming.application\_recipes (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

name TEXT NOT NULL, -- e.g., "JLF Vegetative Growth Foliar Spray", "IMO4 Soil Inoculation Mix"

input\_definition\_id BIGINT REFERENCES gr33n\_natural\_farming.input\_definitions(id) ON DELETE SET NULL, -- General input type this recipe is for

-- OR, if recipe uses specific batches (less common for a general recipe definition, more for a specific application event)

-- input\_batch\_id BIGINT REFERENCES gr33n\_natural\_farming.input\_batches(id) ON DELETE SET NULL,

description TEXT, -- Purpose of this recipe, e.g., "Boost vegetative growth", "Improve soil microbial activity"

target\_application\_type gr33n\_natural\_farming.application\_target\_enum NOT NULL,

dilution\_ratio TEXT, -- e.g., "1:500 (Input:Water)", "2 tbsp per gallon"

components JSONB DEFAULT '{}'::jsonb, -- For multi-input recipes, e.g., [{"input\_definition\_id": 1, "ratio\_part": 1}, {"input\_definition\_id": 5, "ratio\_part": 0.5}, {"carrier": "water", "ratio\_part": 500}]

instructions TEXT, -- Mixing and application instructions

frequency\_guidelines TEXT, -- e.g., "Weekly during vegetative stage", "Once before planting"

target\_crop\_categories gr33n\_crops.crop\_category\_enum[], -- Optional: categories this recipe is good for

target\_growth\_stages TEXT[], -- Optional: text array of growth stages this is good for (e.g., "Seedling", "Fruiting")

notes TEXT,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_application\_recipe\_farm\_name UNIQUE (farm\_id, name, deleted\_at)

);

COMMENT ON TABLE gr33n\_natural\_farming.application\_recipes IS 'Defines standard recipes for diluting and applying natural farming inputs for various purposes.';

-- Link Application Recipes to specific Input Definitions (Many-to-Many if a recipe uses multiple defined inputs)

CREATE TABLE gr33n\_natural\_farming.recipe\_input\_components (

application\_recipe\_id BIGINT NOT NULL REFERENCES gr33n\_natural\_farming.application\_recipes(id) ON DELETE CASCADE,

input\_definition\_id BIGINT NOT NULL REFERENCES gr33n\_natural\_farming.input\_definitions(id) ON DELETE CASCADE,

-- Or input\_batch\_id if a recipe variant uses a specific batch

-- input\_batch\_id BIGINT REFERENCES gr33n\_natural\_farming.input\_batches(id) ON DELETE CASCADE,

part\_value NUMERIC(10,3) NOT NULL, -- e.g., 1 part of this input

part\_unit TEXT DEFAULT 'unitless\_ratio\_part' NOT NULL, -- e.g., 'ml', 'gram', 'unitless\_ratio\_part'

notes TEXT,

PRIMARY KEY (application\_recipe\_id, input\_definition\_id) -- Or include batch\_id if used

-- CONSTRAINT chk\_recipe\_component\_source CHECK (input\_definition\_id IS NOT NULL OR input\_batch\_id IS NOT NULL)

);

COMMENT ON TABLE gr33n\_natural\_farming.recipe\_input\_components IS 'Details the specific natural farming inputs and their proportions within an application recipe.';

-- This table already exists in gr33n\_crops, but we need to ensure it can link to input\_batches from this schema.

-- The `crop\_treatments` table has `natural\_input\_batch\_id BIGINT`. We will add the FK constraint later.

-- ALTER TABLE gr33n\_crops.crop\_treatments

-- ADD CONSTRAINT fk\_crop\_treatment\_natural\_input\_batch

-- FOREIGN KEY (natural\_input\_batch\_id)

-- REFERENCES gr33n\_natural\_farming.input\_batches(id) ON DELETE SET NULL;

--

-- Also, `crop\_treatments` has `inventory\_item\_id`. If base ingredients for natural inputs are in inventory:

-- ALTER TABLE gr33n\_natural\_farming.input\_batches

-- ADD COLUMN base\_ingredient\_inventory\_item\_id BIGINT REFERENCES gr33n\_core.inventory\_items(id) ON DELETE SET NULL;

-- (This is if you track say, 50kg bags of rice bran as an inventory item, then use some for an IMO batch)

--------------------------------------------------------------------------------

-- INDEXES for gr33n\_natural\_farming

--------------------------------------------------------------------------------

-- input\_definitions

CREATE INDEX IF NOT EXISTS idx\_input\_definitions\_farm\_category ON gr33n\_natural\_farming.input\_definitions(farm\_id, category);

CREATE INDEX IF NOT EXISTS idx\_input\_definitions\_deleted\_at ON gr33n\_natural\_farming.input\_definitions(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- input\_batches

CREATE INDEX IF NOT EXISTS idx\_input\_batches\_farm\_definition ON gr33n\_natural\_farming.input\_batches(farm\_id, input\_definition\_id);

CREATE INDEX IF NOT EXISTS idx\_input\_batches\_farm\_status ON gr33n\_natural\_farming.input\_batches(farm\_id, status);

CREATE INDEX IF NOT EXISTS idx\_input\_batches\_ready\_date ON gr33n\_natural\_farming.input\_batches(actual\_ready\_date);

CREATE INDEX IF NOT EXISTS idx\_input\_batches\_deleted\_at ON gr33n\_natural\_farming.input\_batches(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- application\_recipes

CREATE INDEX IF NOT EXISTS idx\_application\_recipes\_farm\_input ON gr33n\_natural\_farming.application\_recipes(farm\_id, input\_definition\_id);

CREATE INDEX IF NOT EXISTS idx\_application\_recipes\_farm\_target\_type ON gr33n\_natural\_farming.application\_recipes(farm\_id, target\_application\_type);

CREATE INDEX IF NOT EXISTS idx\_application\_recipes\_deleted\_at ON gr33n\_natural\_farming.application\_recipes(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- recipe\_input\_components

CREATE INDEX IF NOT EXISTS idx\_recipe\_input\_components\_definition ON gr33n\_natural\_farming.recipe\_input\_components(input\_definition\_id);

--------------------------------------------------------------------------------

-- SCHEMA: gr33n\_crops - Crop cultivation, monitoring, and harvesting

--------------------------------------------------------------------------------

CREATE SCHEMA IF NOT EXISTS gr33n\_crops;

COMMENT ON SCHEMA gr33n\_crops IS 'Manages detailed information about crop varieties, specific plantings/cycles, growth monitoring, pest/disease control, treatments, and harvests. Links to gr33n\_core for farm, zone, user context, and tasks.';

--------------------------------------------------------------------------------

-- ENUMERATED TYPES for gr33n\_crops

--------------------------------------------------------------------------------

CREATE TYPE gr33n\_crops.crop\_category\_enum AS ENUM (

'vegetable\_fruit', 'vegetable\_leafy', 'vegetable\_root', 'vegetable\_legume',

'grain\_cereal', 'grain\_pulse',

'herb\_culinary', 'herb\_medicinal',

'flower\_ornamental', 'flower\_edible',

'cover\_crop', 'fodder\_crop', 'fiber\_crop', 'oilseed\_crop',

'tree\_fruit', 'tree\_nut', 'vine\_fruit',

'mushroom', 'other'

);

COMMENT ON TYPE gr33n\_crops.crop\_category\_enum IS 'Broad categorization of crop types.';

CREATE TYPE gr33n\_crops.plant\_source\_type\_enum AS ENUM (

'seed\_direct\_sown', 'seed\_started\_indoor', 'transplant\_purchased',

'cutting', 'graft', 'bulb\_tuber\_rhizome', 'spore', 'tissue\_culture', 'volunteer'

);

COMMENT ON TYPE gr33n\_crops.plant\_source\_type\_enum IS 'Origin of the plant material for a planting.';

CREATE TYPE gr33n\_crops.observation\_type\_enum AS ENUM (

'general\_note', 'growth\_measurement', 'phenology\_event', -- (e.g., flowering, fruiting)

'water\_stress', 'nutrient\_deficiency', 'pest\_damage\_symptom', 'disease\_symptom',

'environmental\_stress', 'photo\_log', 'soil\_condition\_note', 'pollination\_activity'

);

COMMENT ON TYPE gr33n\_crops.observation\_type\_enum IS 'Type of observation recorded for a crop planting.';

CREATE TYPE gr33n\_crops.treatment\_type\_enum AS ENUM (

'fertilizer\_organic', 'fertilizer\_synthetic', 'soil\_amendment',

'pesticide\_organic', 'pesticide\_synthetic', 'fungicide\_organic', 'fungicide\_synthetic',

'herbicide\_organic', 'herbicide\_synthetic', 'biological\_control\_agent',

'natural\_farming\_input', -- (e.g., JLF, OHN from gr33n\_natural\_farming)

'growth\_regulator', 'physical\_control', -- (e.g., row covers, traps)

'water\_application\_foliar', 'other'

);

COMMENT ON TYPE gr33n\_crops.treatment\_type\_enum IS 'Type of treatment applied to a crop planting.';

CREATE TYPE gr33n\_crops.harvest\_quality\_enum AS ENUM (

'premium\_grade\_a', 'good\_grade\_b', 'fair\_grade\_c', 'processing\_grade', 'cull\_unmarketable', 'feed\_grade'

);

COMMENT ON TYPE gr33n\_crops.harvest\_quality\_enum IS 'Quality assessment of harvested produce.';

--------------------------------------------------------------------------------

-- TABLES for gr33n\_crops

--------------------------------------------------------------------------------

-- Crop Varieties: Master list of crops and their specific varieties a farm might grow.

CREATE TABLE gr33n\_crops.crop\_varieties (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

name TEXT NOT NULL, -- e.g., "Tomato Roma VF", "Carrot Nantes 2"

common\_crop\_name TEXT, -- e.g., "Tomato", "Carrot"

scientific\_name TEXT, -- e.g., "Solanum lycopersicum", "Daucus carota"

category gr33n\_crops.crop\_category\_enum,

description TEXT,

avg\_days\_to\_maturity INTEGER, -- Average days from planting to first harvest

sun\_requirements TEXT, -- e.g., "Full Sun", "Partial Shade"

water\_requirements TEXT, -- e.g., "Moderate", "Drought Tolerant"

soil\_preferences TEXT, -- e.g., "Well-drained loamy", "pH 6.0-6.8"

planting\_instructions TEXT,

pest\_disease\_resistance\_notes TEXT,

characteristics JSONB DEFAULT '{}'::jsonb, -- e.g., {"color": "red", "shape": "plum", "height\_cm": 150, "fruit\_size\_grams": 80}

source\_supplier\_info TEXT, -- Where seeds/starts are typically obtained

is\_active BOOLEAN DEFAULT TRUE NOT NULL, -- If this variety is currently actively used/planned by the farm

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_crop\_variety\_farm\_name UNIQUE (farm\_id, name, deleted\_at) -- Name should be unique per farm unless soft-deleted

);

COMMENT ON TABLE gr33n\_crops.crop\_varieties IS 'Stores details about specific crop varieties grown or planned by a farm. Allows farms to build their own catalog.';

-- Crop Plantings (or Crop Cycles / Batches): Represents a specific instance of a crop being cultivated.

CREATE TABLE gr33n\_crops.crop\_plantings (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Zone where planted. SET NULL if zone is deleted.

crop\_variety\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_varieties(id) ON DELETE RESTRICT, -- Don't delete variety if plantings exist

planting\_name TEXT, -- User-defined name for this specific planting, e.g., "Spring Tomatoes GH1 Bed A", "Field C - Corn 2024"

description TEXT,

planting\_date DATE NOT NULL, -- Date seeds sown or transplants planted

expected\_harvest\_start\_date DATE,

actual\_harvest\_start\_date DATE,

expected\_harvest\_end\_date DATE,

actual\_harvest\_end\_date DATE,

cultivation\_area\_sqm NUMERIC(10,2), -- Area dedicated to this specific planting if different from zone area

plant\_count INTEGER, -- Number of individual plants

plant\_spacing\_cm NUMERIC(6,2),

row\_spacing\_cm NUMERIC(6,2),

source\_type gr33n\_crops.plant\_source\_type\_enum,

source\_details TEXT, -- e.g., "Johnny's Selected Seeds Lot #123", "Local Nursery Batch A"

current\_status TEXT, -- e.g., "Seeding", "Germination", "Vegetative Growth", "Flowering", "Fruiting", "Harvesting", "Completed", "Failed", "Terminated"

notes TEXT,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL

);

COMMENT ON TABLE gr33n\_crops.crop\_plantings IS 'Tracks a specific instance of a crop variety being cultivated, from planting to final harvest or termination.';

-- Standardized Growth Stages (Master list, potentially pre-populated or farm-customizable)

CREATE TABLE gr33n\_crops.crop\_growth\_stages (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE, -- Farm can define custom stages

crop\_category gr33n\_crops.crop\_category\_enum, -- Optional: for general stages per category

crop\_variety\_id BIGINT REFERENCES gr33n\_crops.crop\_varieties(id) ON DELETE CASCADE, -- Optional: for variety-specific stages

stage\_name TEXT NOT NULL, -- e.g., "Seedling (2 true leaves)", "V3 (3 collared leaves)", "Full Bloom", "Fruit Set"

description TEXT,

nominal\_duration\_days INTEGER, -- Typical duration of this stage

sequence\_order INTEGER, -- Order of this stage in the lifecycle

is\_system\_defined BOOLEAN DEFAULT FALSE, -- True if a globally predefined stage

CONSTRAINT uq\_growth\_stage\_farm\_name UNIQUE (farm\_id, stage\_name) WHERE farm\_id IS NOT NULL,

CONSTRAINT uq\_growth\_stage\_system\_name UNIQUE (stage\_name) WHERE farm\_id IS NULL AND crop\_variety\_id IS NULL AND crop\_category IS NULL

);

COMMENT ON TABLE gr33n\_crops.crop\_growth\_stages IS 'Defines standardized or custom growth stages for crops, aiding in consistent tracking.';

-- Planting Growth Stage Log: Tracks when a specific planting enters a growth stage.

CREATE TABLE gr33n\_crops.planting\_growth\_stage\_log (

id BIGSERIAL PRIMARY KEY,

crop\_planting\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_plantings(id) ON DELETE CASCADE,

growth\_stage\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_growth\_stages(id) ON DELETE RESTRICT,

achieved\_date DATE NOT NULL,

notes TEXT,

recorded\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL

);

COMMENT ON TABLE gr33n\_crops.planting\_growth\_stage\_log IS 'Logs when a specific crop planting achieves a defined growth stage.';

-- Crop Observations: General observations, measurements, or photo logs for a planting.

CREATE TABLE gr33n\_crops.crop\_observations (

id BIGSERIAL PRIMARY KEY,

crop\_planting\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_plantings(id) ON DELETE CASCADE,

observation\_time TIMESTAMPTZ DEFAULT NOW() NOT NULL,

observation\_type gr33n\_crops.observation\_type\_enum NOT NULL,

description TEXT NOT NULL,

value\_numeric NUMERIC(10,2), -- For quantitative observations like height, count

value\_text TEXT, -- For qualitative data

file\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL, -- For linking photos

related\_sensor\_id BIGINT REFERENCES gr33n\_core.sensors(id) ON DELETE SET NULL, -- If observation is related to a sensor reading

recorded\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL

);

COMMENT ON TABLE gr33n\_crops.crop\_observations IS 'Records various observations, measurements, or media related to a crop planting over time.';

-- Pest & Disease Sightings: Logging specific pest or disease events.

CREATE TABLE gr33n\_crops.pest\_disease\_sightings (

id BIGSERIAL PRIMARY KEY,

crop\_planting\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_plantings(id) ON DELETE CASCADE,

sighting\_time TIMESTAMPTZ DEFAULT NOW() NOT NULL,

pest\_or\_disease\_name TEXT NOT NULL, -- e.g., "Aphids", "Powdery Mildew", "Tomato Hornworm"

type TEXT NOT NULL CHECK (type IN ('pest', 'disease', 'weed')),

severity TEXT, -- e.g., "Low", "Medium", "High", "Localized", "Widespread"

description TEXT, -- Symptoms observed, extent of infestation/infection

affected\_plant\_part TEXT, -- e.g., "Leaves", "Fruit", "Stem", "Roots"

estimated\_affected\_area\_percent NUMERIC(5,2),

file\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL, -- Photo of the issue

recorded\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL

);

COMMENT ON TABLE gr33n\_crops.pest\_disease\_sightings IS 'Logs occurrences and details of pests, diseases, or significant weed pressure affecting a crop planting.';

-- Crop Treatments/Applications: Records of fertilizers, pesticides, amendments applied.

CREATE TABLE gr33n\_crops.crop\_treatments (

id BIGSERIAL PRIMARY KEY,

crop\_planting\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_plantings(id) ON DELETE CASCADE,

pest\_disease\_sighting\_id BIGINT REFERENCES gr33n\_crops.pest\_disease\_sightings(id) ON DELETE SET NULL, -- If treatment is in response to a specific sighting

application\_date DATE NOT NULL,

treatment\_type gr33n\_crops.treatment\_type\_enum NOT NULL,

product\_name TEXT, -- Specific product used, e.g., "Neem Oil Extract", "Roundup", "Miracle-Gro All Purpose"

-- Link to inventory item for product from gr33n\_core.inventory\_items

inventory\_item\_id BIGINT, -- FK to gr33n\_core.inventory\_items (add later if inventory schema is defined)

-- Link to natural farming input batch from gr33n\_natural\_farming.input\_batches

natural\_input\_batch\_id BIGINT, -- FK to gr33n\_natural\_farming.input\_batches (add later)

application\_method TEXT, -- e.g., "Foliar Spray", "Soil Drench", "Broadcast Spreading", "Injection"

concentration TEXT, -- e.g., "10ml/gallon", "As directed"

amount\_applied\_value NUMERIC(10,2),

amount\_applied\_unit TEXT, -- e.g., "gallons", "lbs", "liters", "kg"

area\_treated\_sqm NUMERIC(10,2),

reason\_for\_application TEXT, -- e.g., "Preventative", "Aphid control", "Nutrient boost"

weather\_conditions\_at\_application TEXT, -- e.g., "Cloudy, 15C, light wind"

notes TEXT,

applied\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL

);

COMMENT ON TABLE gr33n\_crops.crop\_treatments IS 'Records applications of fertilizers, pesticides, soil amendments, or other treatments to a crop planting.';

-- TODO: Add FK constraints for inventory\_item\_id and natural\_input\_batch\_id once those schemas/tables are defined.

-- Crop Harvests: Logging harvest events and yields for a planting.

CREATE TABLE gr33n\_crops.crop\_harvests (

id BIGSERIAL PRIMARY KEY,

crop\_planting\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_plantings(id) ON DELETE CASCADE,

harvest\_date DATE NOT NULL,

quantity\_harvested NUMERIC(12,2) NOT NULL,

unit\_of\_measure TEXT NOT NULL, -- e.g., "kg", "lbs", "bushels", "bunches", "pieces"

quality\_grade gr33n\_crops.harvest\_quality\_enum,

storage\_location TEXT, -- Where the harvested produce is stored

market\_destination TEXT, -- e.g., "Farmers Market", "CSA Share", "Wholesale", "On-farm consumption"

notes TEXT,

harvested\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL

);

COMMENT ON TABLE gr33n\_crops.crop\_harvests IS 'Records details of each harvest event from a crop planting, including yield and quality.';

-- Soil Tests: Records of soil analysis results.

CREATE TABLE gr33n\_crops.soil\_tests (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Soil test might be for a specific zone

crop\_planting\_id BIGINT REFERENCES gr33n\_crops.crop\_plantings(id) ON DELETE SET NULL, -- Or related to a specific planting

test\_date DATE NOT NULL,

lab\_name TEXT, -- Name of the laboratory that performed the test

report\_reference TEXT, -- Lab report number or identifier

ph\_value NUMERIC(4,2),

organic\_matter\_percent NUMERIC(5,2),

nitrogen\_ppm NUMERIC(7,2), -- Or other unit like lbs/acre

phosphorus\_ppm NUMERIC(7,2),

potassium\_ppm NUMERIC(7,2),

calcium\_ppm NUMERIC(7,2),

magnesium\_ppm NUMERIC(7,2),

sulfur\_ppm NUMERIC(7,2),

zinc\_ppm NUMERIC(7,2),

iron\_ppm NUMERIC(7,2),

manganese\_ppm NUMERIC(7,2),

copper\_ppm NUMERIC(7,2),

boron\_ppm NUMERIC(7,2),

sodium\_ppm NUMERIC(7,2),

electrical\_conductivity\_ds\_m NUMERIC(6,2), -- dS/m

cation\_exchange\_capacity\_meq\_100g NUMERIC(6,2), -- meq/100g

recommendations TEXT, -- Recommendations from the lab or agronomist

full\_report\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT chk\_soil\_test\_location CHECK (zone\_id IS NOT NULL OR crop\_planting\_id IS NOT NULL) -- Must be linked to at least one

);

COMMENT ON TABLE gr33n\_crops.soil\_tests IS 'Stores results of soil analysis tests, linked to zones or specific crop plantings.';

--------------------------------------------------------------------------------

-- Linking Tables (if needed, many-to-many relationships not immediately obvious here)

--------------------------------------------------------------------------------

-- Example: If a single treatment application could target multiple pest/disease sightings

-- CREATE TABLE gr33n\_crops.treatment\_sighting\_links (

-- crop\_treatment\_id BIGINT NOT NULL REFERENCES gr33n\_crops.crop\_treatments(id) ON DELETE CASCADE,

-- pest\_disease\_sighting\_id BIGINT NOT NULL REFERENCES gr33n\_crops.pest\_disease\_sightings(id) ON DELETE CASCADE,

-- PRIMARY KEY (crop\_treatment\_id, pest\_disease\_sighting\_id)

-- );

-- COMMENT ON TABLE gr33n\_crops.treatment\_sighting\_links IS 'Links a single treatment application to multiple pest/disease sightings it addresses.';

--------------------------------------------------------------------------------

-- SCHEMA: gr33n\_animals - Livestock and animal husbandry management

--------------------------------------------------------------------------------

CREATE SCHEMA IF NOT EXISTS gr33n\_animals;

COMMENT ON SCHEMA gr33n\_animals IS 'Manages information about individual animals, animal groups, breeds, feeding, health records, breeding, movements, and animal-related productions (e.g., eggs, milk, wool).';

--------------------------------------------------------------------------------

-- ENUMERATED TYPES for gr33n\_animals

--------------------------------------------------------------------------------

CREATE TYPE gr33n\_animals.animal\_sex\_enum AS ENUM (

'male', 'female', 'castrated\_male', 'spayed\_female', 'unknown', 'hermaphrodite', 'other'

);

COMMENT ON TYPE gr33n\_animals.animal\_sex\_enum IS 'Sex of an animal.';

CREATE TYPE gr33n\_animals.acquisition\_method\_enum AS ENUM (

'born\_on\_farm', 'purchased', 'hatched\_on\_farm', 'gift\_trade', 'leased\_borrowed', 'other'

);

COMMENT ON TYPE gr33n\_animals.acquisition\_method\_enum IS 'How an animal was acquired by the farm.';

CREATE TYPE gr33n\_animals.disposition\_method\_enum AS ENUM (

'sold\_live', 'sold\_meat', 'processed\_on\_farm', 'culled', 'died\_natural\_causes', 'died\_predation\_accident',

'transferred\_off\_farm', 'leased\_out', 'gifted\_traded\_off', 'escaped\_lost', 'other'

);

COMMENT ON TYPE gr33n\_animals.disposition\_method\_enum IS 'How an animal left the farm or its current status if deceased/gone.';

CREATE TYPE gr33n\_animals.health\_event\_type\_enum AS ENUM (

'vaccination', 'deworming', 'medication\_treatment', 'injury', 'illness\_symptom',

'routine\_checkup', 'parasite\_control', 'quarantine', 'hoof\_trimming', 'grooming',

'diagnostic\_test', 'surgery', 'other'

);

COMMENT ON TYPE gr33n\_animals.health\_event\_type\_enum IS 'Type of health-related event or treatment.';

CREATE TYPE gr33n\_animals.breeding\_event\_type\_enum AS ENUM (

'heat\_observed', 'artificial\_insemination', 'natural\_mating', 'pregnancy\_check\_positive',

'pregnancy\_check\_negative', 'birthing\_hatching', 'weaning', 'abortion\_stillbirth', 'other'

);

COMMENT ON TYPE gr33n\_animals.breeding\_event\_type\_enum IS 'Type of breeding-related event.';

CREATE TYPE gr33n\_animals.production\_type\_enum AS ENUM (

'milk', 'eggs', 'wool\_fiber', 'honey', 'meat\_carcass', 'manure', 'offspring\_live', 'other'

);

COMMENT ON TYPE gr33n\_animals.production\_type\_enum IS 'Type of production harvested or collected from animals.';

--------------------------------------------------------------------------------

-- TABLES for gr33n\_animals

--------------------------------------------------------------------------------

-- Animal Species & Breeds: Catalog of animal species and their specific breeds relevant to the farm.

CREATE TABLE gr33n\_animals.animal\_breeds (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

species\_name TEXT NOT NULL, -- e.g., "Chicken", "Cow", "Goat", "Sheep", "Pig", "Bee"

breed\_name TEXT NOT NULL, -- e.g., "Rhode Island Red", "Holstein", "Nubian", "Dorper", "Yorkshire", "Italian Honey Bee"

common\_names TEXT, -- Other common names for the breed/species

description TEXT, -- Characteristics of the breed

origin\_country TEXT,

conservation\_status TEXT, -- e.g., "Common", "At Risk", "Endangered"

avg\_lifespan\_years NUMERIC(5,1),

avg\_mature\_weight\_kg NUMERIC(7,2),

primary\_purpose TEXT, -- e.g., "Eggs", "Milk", "Meat", "Fiber", "Pollination", "Draft"

notes TEXT,

is\_active BOOLEAN DEFAULT TRUE NOT NULL, -- If this breed is actively managed by the farm

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_animal\_breed\_farm\_species\_name UNIQUE (farm\_id, species\_name, breed\_name, deleted\_at)

);

COMMENT ON TABLE gr33n\_animals.animal\_breeds IS 'Stores information about animal species and specific breeds relevant to the farm.';

-- Animal Individuals: Detailed records for each individually managed animal.

CREATE TABLE gr33n\_animals.animal\_individuals (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

animal\_breed\_id BIGINT NOT NULL REFERENCES gr33n\_animals.animal\_breeds(id) ON DELETE RESTRICT,

tag\_id TEXT UNIQUE, -- Ear tag, leg band, RFID, microchip ID. Could be unique per farm instead of globally.

name TEXT, -- Common name given to the animal

sex gr33n\_animals.animal\_sex\_enum,

birth\_date DATE,

birth\_weight\_kg NUMERIC(7,2),

dam\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE SET NULL, -- Mother

sire\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE SET NULL, -- Father

acquisition\_date DATE,

acquisition\_method gr33n\_animals.acquisition\_method\_enum,

source\_details TEXT, -- e.g., "Purchased from Smith Farms", "Hatchery Batch #456"

current\_zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL, -- Current pasture, pen, barn

disposition\_date DATE,

disposition\_method gr33n\_animals.disposition\_method\_enum,

disposition\_reason TEXT,

notes TEXT,

profile\_image\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_animal\_farm\_tag\_id UNIQUE (farm\_id, tag\_id, deleted\_at) WHERE tag\_id IS NOT NULL -- Tag ID unique per farm

);

COMMENT ON TABLE gr33n\_animals.animal\_individuals IS 'Tracks individual animals with unique identifiers, parentage, and lifecycle events.';

-- Animal Groups: For managing groups or herds of animals where individual tracking isn't primary (e.g., flock of chickens, beehive colony).

CREATE TABLE gr33n\_animals.animal\_groups (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

animal\_breed\_id BIGINT NOT NULL REFERENCES gr33n\_animals.animal\_breeds(id) ON DELETE RESTRICT,

group\_name TEXT NOT NULL, -- e.g., "Layer Flock A", "Main Pasture Beef Herd", "Apiary Hive #3"

description TEXT,

current\_zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL,

establishment\_date DATE,

initial\_count INTEGER,

current\_approx\_count INTEGER,

avg\_age\_months NUMERIC(5,1),

primary\_purpose TEXT, -- e.g., "Egg Production", "Grazing", "Pollination"

notes TEXT,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT uq\_animal\_group\_farm\_name UNIQUE (farm\_id, group\_name, deleted\_at)

);

COMMENT ON TABLE gr33n\_animals.animal\_groups IS 'Manages groups, flocks, herds, or colonies of animals, especially where individual tracking is not primary.';

-- Link table for individuals within groups (an animal can be in one group at a time for simplicity here, or this can be time-bound)

CREATE TABLE gr33n\_animals.animal\_group\_members (

animal\_individual\_id BIGINT NOT NULL REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE,

animal\_group\_id BIGINT NOT NULL REFERENCES gr33n\_animals.animal\_groups(id) ON DELETE CASCADE,

date\_joined DATE DEFAULT CURRENT\_DATE NOT NULL,

date\_left DATE,

reason\_for\_leaving TEXT,

PRIMARY KEY (animal\_individual\_id, animal\_group\_id, date\_joined) -- An animal could rejoin a group later

-- Could add a constraint: an animal can only be in one ACTIVE group at a time if date\_left IS NULL.

);

COMMENT ON TABLE gr33n\_animals.animal\_group\_members IS 'Associates individual animals with specific animal groups over time.';

-- Feeding Records: Logs feeding events for individuals or groups.

CREATE TABLE gr33n\_animals.feeding\_records (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

animal\_individual\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE,

animal\_group\_id BIGINT REFERENCES gr33n\_animals.animal\_groups(id) ON DELETE CASCADE,

feed\_time TIMESTAMPTZ DEFAULT NOW() NOT NULL,

feed\_type\_name TEXT NOT NULL, -- e.g., "Layer Pellets", "Pasture Grass", "Hay", "Custom Grain Mix"

-- Link to inventory item for feed from gr33n\_core.inventory\_items

feed\_inventory\_item\_id BIGINT, -- FK to gr33n\_core.inventory\_items (add later)

quantity\_fed NUMERIC(10,2) NOT NULL,

quantity\_unit TEXT NOT NULL, -- e.g., "kg", "lbs", "scoops", "bales"

notes TEXT,

fed\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

-- No updated\_at/deleted\_at for simple log records usually. Updated if editable.

CONSTRAINT chk\_feeding\_target CHECK (animal\_individual\_id IS NOT NULL OR animal\_group\_id IS NOT NULL)

);

COMMENT ON TABLE gr33n\_animals.feeding\_records IS 'Logs feeding events, detailing feed type and quantity for individual animals or groups.';

-- TODO: Add FK constraint for feed\_inventory\_item\_id once gr33n\_core.inventory\_items is finalized.

-- Health Records: Tracks health events, treatments, vaccinations.

CREATE TABLE gr33n\_animals.health\_records (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

animal\_individual\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE,

animal\_group\_id BIGINT REFERENCES gr33n\_animals.animal\_groups(id) ON DELETE CASCADE, -- For flock/herd treatments

event\_date DATE NOT NULL,

event\_type gr33n\_animals.health\_event\_type\_enum NOT NULL,

description TEXT, -- Specific illness, injury details, medication name, vaccine batch

product\_used TEXT, -- e.g., "Ivermectin Pour-On", "Penicillin G", "Bovishield Gold FP5"

-- Link to inventory item for medication/product from gr33n\_core.inventory\_items

product\_inventory\_item\_id BIGINT, -- FK to gr33n\_core.inventory\_items (add later)

dosage\_administered TEXT, -- e.g., "10ml", "1 pill", "0.5cc/10lbs"

withdrawal\_period\_days INTEGER, -- If applicable for meat/milk/eggs

veterinarian\_involved TEXT,

diagnosis TEXT,

outcome TEXT, -- e.g., "Recovered", "Ongoing", "Died", "Culled"

notes TEXT,

recorded\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

file\_attachment\_id BIGINT REFERENCES gr33n\_core.file\_attachments(id) ON DELETE SET NULL, -- Vet reports, lab results

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT chk\_health\_record\_target CHECK (animal\_individual\_id IS NOT NULL OR animal\_group\_id IS NOT NULL)

);

COMMENT ON TABLE gr33n\_animals.health\_records IS 'Tracks health-related events, treatments, and diagnoses for individual animals or groups.';

-- TODO: Add FK constraint for product\_inventory\_item\_id.

-- Breeding Records: Tracks mating, pregnancy, birthing/hatching events.

CREATE TABLE gr33n\_animals.breeding\_records (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

female\_animal\_id BIGINT NOT NULL REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE, -- Dam

male\_animal\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE SET NULL, -- Sire (can be null for AI or unknown sire in a group)

animal\_group\_id BIGINT REFERENCES gr33n\_animals.animal\_groups(id) ON DELETE SET NULL, -- If mating occurred within a group

event\_date DATE NOT NULL,

event\_type gr33n\_animals.breeding\_event\_type\_enum NOT NULL,

semen\_id\_or\_source TEXT, -- For AI, e.g., bull ID, semen batch #

expected\_due\_date DATE,

actual\_birth\_hatch\_date DATE,

number\_of\_offspring INTEGER,

number\_of\_live\_offspring INTEGER,

number\_of\_male\_offspring INTEGER,

number\_of\_female\_offspring INTEGER,

complications TEXT,

notes TEXT,

recorded\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL

);

COMMENT ON TABLE gr33n\_animals.breeding\_records IS 'Tracks breeding events, pregnancies, and birthing/hatching outcomes.';

-- Offspring Links: Links offspring recorded in animal\_individuals back to a specific breeding\_record.

CREATE TABLE gr33n\_animals.breeding\_offspring (

breeding\_record\_id BIGINT NOT NULL REFERENCES gr33n\_animals.breeding\_records(id) ON DELETE CASCADE,

offspring\_animal\_id BIGINT NOT NULL REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE,

notes TEXT,

PRIMARY KEY (breeding\_record\_id, offspring\_animal\_id)

);

COMMENT ON TABLE gr33n\_animals.breeding\_offspring IS 'Links individual offspring records to their parent breeding event record.';

-- Animal Movements: Tracks movement of individuals or groups between zones.

CREATE TABLE gr33n\_animals.animal\_movements (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

animal\_individual\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE,

animal\_group\_id BIGINT REFERENCES gr33n\_animals.animal\_groups(id) ON DELETE CASCADE,

movement\_date TIMESTAMPTZ DEFAULT NOW() NOT NULL,

from\_zone\_id BIGINT REFERENCES gr33n\_core.zones(id) ON DELETE SET NULL,

to\_zone\_id BIGINT NOT NULL REFERENCES gr33n\_core.zones(id) ON DELETE RESTRICT,

reason TEXT,

notes TEXT,

moved\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

-- No updated\_at/deleted\_at for simple log records.

CONSTRAINT chk\_movement\_target CHECK (animal\_individual\_id IS NOT NULL OR animal\_group\_id IS NOT NULL)

);

COMMENT ON TABLE gr33n\_animals.animal\_movements IS 'Logs the movement of animals or groups between different zones on the farm.';

-- Production Records: Logs collection of milk, eggs, wool, honey, etc.

CREATE TABLE gr33n\_animals.production\_records (

id BIGSERIAL PRIMARY KEY,

farm\_id BIGINT NOT NULL REFERENCES gr33n\_core.farms(id) ON DELETE CASCADE,

animal\_individual\_id BIGINT REFERENCES gr33n\_animals.animal\_individuals(id) ON DELETE CASCADE,

animal\_group\_id BIGINT REFERENCES gr33n\_animals.animal\_groups(id) ON DELETE CASCADE, -- e.g., for egg collection from a flock

production\_date DATE NOT NULL,

production\_type gr33n\_animals.production\_type\_enum NOT NULL,

quantity\_produced NUMERIC(10,2) NOT NULL,

quantity\_unit TEXT NOT NULL, -- e.g., "liters", "gallons", "dozen", "pieces", "kg", "lbs"

quality\_grade TEXT, -- e.g., "Grade A Large", "Extra Fancy"

storage\_location TEXT,

notes TEXT,

collected\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

related\_task\_id BIGINT REFERENCES gr33n\_core.tasks(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_at TIMESTAMPTZ DEFAULT NOW() NOT NULL,

updated\_by\_user\_id UUID REFERENCES gr33n\_core.profiles(user\_id) ON DELETE SET NULL,

deleted\_at TIMESTAMPTZ DEFAULT NULL,

CONSTRAINT chk\_production\_target CHECK (animal\_individual\_id IS NOT NULL OR animal\_group\_id IS NOT NULL)

);

COMMENT ON TABLE gr33n\_animals.production\_records IS 'Records collection of animal products like milk, eggs, wool, honey.';

--------------------------------------------------------------------------------

-- INDEXES for gr33n\_animals

--------------------------------------------------------------------------------

-- animal\_breeds

CREATE INDEX IF NOT EXISTS idx\_animal\_breeds\_farm\_species ON gr33n\_animals.animal\_breeds(farm\_id, species\_name);

CREATE INDEX IF NOT EXISTS idx\_animal\_breeds\_deleted\_at ON gr33n\_animals.animal\_breeds(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- animal\_individuals

CREATE INDEX IF NOT EXISTS idx\_animal\_individuals\_farm\_breed ON gr33n\_animals.animal\_individuals(farm\_id, animal\_breed\_id);

CREATE INDEX IF NOT EXISTS idx\_animal\_individuals\_farm\_sex ON gr33n\_animals.animal\_individuals(farm\_id, sex);

CREATE INDEX IF NOT EXISTS idx\_animal\_individuals\_current\_zone ON gr33n\_animals.animal\_individuals(current\_zone\_id);

CREATE INDEX IF NOT EXISTS idx\_animal\_individuals\_dam\_id ON gr33n\_animals.animal\_individuals(dam\_id);

CREATE INDEX IF NOT EXISTS idx\_animal\_individuals\_sire\_id ON gr33n\_animals.animal\_individuals(sire\_id);

CREATE INDEX IF NOT EXISTS idx\_animal\_individuals\_deleted\_at ON gr33n\_animals.animal\_individuals(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- animal\_groups

CREATE INDEX IF NOT EXISTS idx\_animal\_groups\_farm\_breed ON gr33n\_animals.animal\_groups(farm\_id, animal\_breed\_id);

CREATE INDEX IF NOT EXISTS idx\_animal\_groups\_current\_zone ON gr33n\_animals.animal\_groups(current\_zone\_id);

CREATE INDEX IF NOT EXISTS idx\_animal\_groups\_deleted\_at ON gr33n\_animals.animal\_groups(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- animal\_group\_members

CREATE INDEX IF NOT EXISTS idx\_animal\_group\_members\_group\_id ON gr33n\_animals.animal\_group\_members(animal\_group\_id);

-- feeding\_records

CREATE INDEX IF NOT EXISTS idx\_feeding\_records\_farm\_individual ON gr33n\_animals.feeding\_records(farm\_id, animal\_individual\_id, feed\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_feeding\_records\_farm\_group ON gr33n\_animals.feeding\_records(farm\_id, animal\_group\_id, feed\_time DESC);

CREATE INDEX IF NOT EXISTS idx\_feeding\_records\_feed\_type ON gr33n\_animals.feeding\_records(farm\_id, feed\_type\_name);

-- health\_records

CREATE INDEX IF NOT EXISTS idx\_health\_records\_farm\_individual ON gr33n\_animals.health\_records(farm\_id, animal\_individual\_id, event\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_health\_records\_farm\_group ON gr33n\_animals.health\_records(farm\_id, animal\_group\_id, event\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_health\_records\_event\_type ON gr33n\_animals.health\_records(farm\_id, event\_type);

CREATE INDEX IF NOT EXISTS idx\_health\_records\_deleted\_at ON gr33n\_animals.health\_records(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- breeding\_records

CREATE INDEX IF NOT EXISTS idx\_breeding\_records\_farm\_female ON gr33n\_animals.breeding\_records(farm\_id, female\_animal\_id, event\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_breeding\_records\_farm\_male ON gr33n\_animals.breeding\_records(farm\_id, male\_animal\_id);

CREATE INDEX IF NOT EXISTS idx\_breeding\_records\_event\_type ON gr33n\_animals.breeding\_records(farm\_id, event\_type);

CREATE INDEX IF NOT EXISTS idx\_breeding\_records\_deleted\_at ON gr33n\_animals.breeding\_records(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;

-- breeding\_offspring

CREATE INDEX IF NOT EXISTS idx\_breeding\_offspring\_offspring\_id ON gr33n\_animals.breeding\_offspring(offspring\_animal\_id);

-- animal\_movements

CREATE INDEX IF NOT EXISTS idx\_animal\_movements\_farm\_individual ON gr33n\_animals.animal\_movements(farm\_id, animal\_individual\_id, movement\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_animal\_movements\_farm\_group ON gr33n\_animals.animal\_movements(farm\_id, animal\_group\_id, movement\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_animal\_movements\_to\_zone ON gr33n\_animals.animal\_movements(to\_zone\_id);

-- production\_records

CREATE INDEX IF NOT EXISTS idx\_production\_records\_farm\_individual ON gr33n\_animals.production\_records(farm\_id, animal\_individual\_id, production\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_production\_records\_farm\_group ON gr33n\_animals.production\_records(farm\_id, animal\_group\_id, production\_date DESC);

CREATE INDEX IF NOT EXISTS idx\_production\_records\_type ON gr33n\_animals.production\_records(farm\_id, production\_type);

CREATE INDEX IF NOT EXISTS idx\_production\_records\_deleted\_at ON gr33n\_animals.production\_records(deleted\_at NULLS FIRST) WHERE deleted\_at IS NULL;