**04 - Data Modellen voor NEXA**

**Status:** IN PROGRESS

**Inhoudsopgave:**

1. [Supabase (PostgreSQL) Datamodel](#1-supabase-postgresql-datamodel)
2. [Neo4j Graph Model](#2-neo4j-graph-model)

**1. Supabase (PostgreSQL) Datamodel**

Dit beschrijft het gedetailleerde schema voor de PostgreSQL database beheerd door Supabase. RLS policies worden apart gedefinieerd (zie Security Architectuur).

**-- ### Authenticatie & Autorisatie Gerelateerd ###**

-- `users` tabel wordt beheerd door Supabase Auth, hier geen aparte DDL nodig.

-- `audit\_log\_entries` tabel wordt beheerd door Supabase Auth, hier geen aparte DDL nodig.

**-- Organisaties (voor multi-tenancy/teams)**

CREATE TABLE organizations (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

name TEXT NOT NULL,

owner\_id uuid REFERENCES auth.users(id) ON DELETE SET NULL, -- Wie heeft de org aangemaakt (optioneel)

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

ALTER TABLE organizations ENABLE ROW LEVEL SECURITY; -- Belangrijk voor data scheiding

**-- Koppeling gebruikers aan organisaties**

CREATE TABLE organization\_members (

organization\_id uuid REFERENCES organizations(id) ON DELETE CASCADE NOT NULL,

user\_id uuid REFERENCES auth.users(id) ON DELETE CASCADE NOT NULL,

role TEXT NOT NULL DEFAULT 'member', -- bijv. 'admin', 'member'

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

PRIMARY KEY (organization\_id, user\_id) -- Een gebruiker kan maar 1 rol hebben per org

);

ALTER TABLE organization\_members ENABLE ROW LEVEL SECURITY;

**-- ### Salesforce Connectie & Snapshot Beheer ###**

CREATE TABLE salesforce\_connections (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

organization\_id uuid REFERENCES organizations(id) ON DELETE CASCADE NOT NULL,

sf\_org\_id TEXT NOT NULL UNIQUE, -- Salesforce Org ID (18-char), uniek over alle connecties

sf\_user\_id TEXT, -- Salesforce User ID van authenticerende gebruiker

instance\_url TEXT NOT NULL,

-- Tokens worden versleuteld in de applicatielaag VOOR opslag

access\_token\_encrypted TEXT,

refresh\_token\_encrypted TEXT,

status TEXT NOT NULL DEFAULT 'inactive', -- 'active', 'inactive', 'revoked', 'needs\_reauth'

last\_connected\_at TIMESTAMPTZ,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

user\_id uuid REFERENCES auth.users(id) ON DELETE SET NULL -- Wie de koppeling legde

);

CREATE INDEX idx\_sf\_connections\_org\_id ON salesforce\_connections(organization\_id);

ALTER TABLE salesforce\_connections ENABLE ROW LEVEL SECURITY;

CREATE TABLE metadata\_snapshots (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

sf\_connection\_id uuid REFERENCES salesforce\_connections(id) ON DELETE CASCADE NOT NULL,

status TEXT NOT NULL DEFAULT 'PENDING', -- 'PENDING', 'FETCHING', ..., 'COMPLETED', 'FAILED'

error\_message TEXT,

triggered\_by uuid REFERENCES auth.users(id) ON DELETE SET NULL,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

completed\_at TIMESTAMPTZ

);

CREATE INDEX idx\_snapshots\_sf\_connection\_id ON metadata\_snapshots(sf\_connection\_id);

CREATE INDEX idx\_snapshots\_status ON metadata\_snapshots(status);

ALTER TABLE metadata\_snapshots ENABLE ROW LEVEL SECURITY;

**-- ### Geparsed Metadata Opslag ###**

**-- Deze tabellen bevatten de 'ruwe' geparsede data per snapshot.**

CREATE TABLE parsed\_components (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

snapshot\_id uuid REFERENCES metadata\_snapshots(id) ON DELETE CASCADE NOT NULL,

component\_type TEXT NOT NULL, -- 'CustomObject', 'CustomField', 'Flow', 'ApexTrigger', 'Report', etc.

api\_name TEXT NOT NULL, -- Volledige API naam

label TEXT,

sf\_id TEXT, -- Salesforce ID (indien beschikbaar)

-- Type-specifieke attributen, flexibel opgeslagen

attributes JSONB DEFAULT '{}'::jsonb,

-- Optioneel: Ruwe definitie voor diepere analyse of debuggen

-- raw\_definition TEXT,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

-- Zorgt voor uniciteit van een component binnen een snapshot

UNIQUE (snapshot\_id, api\_name)

);

CREATE INDEX idx\_components\_snapshot\_id ON parsed\_components(snapshot\_id);

CREATE INDEX idx\_components\_snapshot\_type ON parsed\_components(snapshot\_id, component\_type);

CREATE INDEX idx\_components\_snapshot\_apiname ON parsed\_components(snapshot\_id, api\_name); -- Al gedekt door UNIQUE

ALTER TABLE parsed\_components ENABLE ROW LEVEL SECURITY;

-- Expliciete relaties TUSSEN componenten zoals ontdekt tijdens parsing

CREATE TABLE parsed\_relationships (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

snapshot\_id uuid REFERENCES metadata\_snapshots(id) ON DELETE CASCADE NOT NULL,

-- We slaan api\_names op ipv foreign keys naar parsed\_components

-- Dit maakt het makkelijker om relaties te definieren tijdens parsing,

-- voordat alle component IDs bekend zijn, en maakt lookups tijdens graph

-- populatie mogelijk direct op api\_name (wat ook in Neo4j gebruikt wordt).

source\_component\_api\_name TEXT NOT NULL,

source\_component\_type TEXT NOT NULL,

target\_component\_api\_name TEXT NOT NULL,

target\_component\_type TEXT NOT NULL,

relationship\_type TEXT NOT NULL, -- 'FIELD\_USED\_IN\_FLOW', 'FLOW\_REFERENCES\_FIELD', etc.

context JSONB, -- Extra context, bijv. specifiek flow element

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

CREATE INDEX idx\_relationships\_snapshot\_id ON parsed\_relationships(snapshot\_id);

CREATE INDEX idx\_relationships\_snapshot\_source ON parsed\_relationships(snapshot\_id, source\_component\_api\_name);

CREATE INDEX idx\_relationships\_snapshot\_target ON parsed\_relationships(snapshot\_id, target\_component\_api\_name);

CREATE INDEX idx\_relationships\_snapshot\_type ON parsed\_relationships(snapshot\_id, relationship\_type);

ALTER TABLE parsed\_relationships ENABLE ROW LEVEL SECURITY;

**-- ### Analyse & AI Resultaten ###**

CREATE TABLE analysis\_results (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

snapshot\_id uuid REFERENCES metadata\_snapshots(id) ON DELETE CASCADE NOT NULL,

analysis\_type TEXT NOT NULL, -- 'Flow Inconsistency Check', 'Unused Component', etc.

-- Verwijzing naar het hoofdcomponent waar het resultaat over gaat

component\_api\_name TEXT,

component\_type TEXT,

severity TEXT NOT NULL DEFAULT 'Info', -- 'Error', 'Warning', 'Info'

description TEXT NOT NULL,

details JSONB, -- Gestructureerde details over het probleem/resultaat

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

CREATE INDEX idx\_analysis\_results\_snapshot\_id ON analysis\_results(snapshot\_id);

CREATE INDEX idx\_analysis\_results\_snapshot\_type ON analysis\_results(snapshot\_id, analysis\_type);

CREATE INDEX idx\_analysis\_results\_snapshot\_component ON analysis\_results(snapshot\_id, component\_api\_name);

ALTER TABLE analysis\_results ENABLE ROW LEVEL SECURITY;

CREATE TABLE ai\_interactions (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

snapshot\_id uuid REFERENCES metadata\_snapshots(id) ON DELETE SET NULL, -- Koppeling aan snapshot (kan null zijn?)

user\_id uuid REFERENCES auth.users(id) ON DELETE SET NULL,

prompt TEXT NOT NULL,

neo4j\_query\_summary TEXT,

llm\_response TEXT,

feedback\_rating INT CHECK (feedback\_rating IN (-1, 1)), -- -1 = down, 1 = up

feedback\_comment TEXT,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

CREATE INDEX idx\_ai\_interactions\_user\_id ON ai\_interactions(user\_id);

ALTER TABLE ai\_interactions ENABLE ROW LEVEL SECURITY;

**-- ### Analyse & AI Resultaten ###**

CREATE TABLE analysis\_results (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

snapshot\_id uuid REFERENCES metadata\_snapshots(id) ON DELETE CASCADE NOT NULL,

analysis\_type TEXT NOT NULL, -- 'Flow Inconsistency Check', 'Unused Component', etc.

-- Verwijzing naar het hoofdcomponent waar het resultaat over gaat

component\_api\_name TEXT,

component\_type TEXT,

severity TEXT NOT NULL DEFAULT 'Info', -- 'Error', 'Warning', 'Info'

description TEXT NOT NULL,

details JSONB, -- Gestructureerde details over het probleem/resultaat

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

CREATE INDEX idx\_analysis\_results\_snapshot\_id ON analysis\_results(snapshot\_id);

CREATE INDEX idx\_analysis\_results\_snapshot\_type ON analysis\_results(snapshot\_id, analysis\_type);

CREATE INDEX idx\_analysis\_results\_snapshot\_component ON analysis\_results(snapshot\_id, component\_api\_name);

ALTER TABLE analysis\_results ENABLE ROW LEVEL SECURITY;

CREATE TABLE ai\_interactions (

id uuid PRIMARY KEY DEFAULT uuid\_generate\_v4(),

snapshot\_id uuid REFERENCES metadata\_snapshots(id) ON DELETE SET NULL, -- Koppeling aan snapshot (kan null zijn?)

user\_id uuid REFERENCES auth.users(id) ON DELETE SET NULL,

prompt TEXT NOT NULL,

neo4j\_query\_summary TEXT,

llm\_response TEXT,

feedback\_rating INT CHECK (feedback\_rating IN (-1, 1)), -- -1 = down, 1 = up

feedback\_comment TEXT,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

CREATE INDEX idx\_ai\_interactions\_user\_id ON ai\_interactions(user\_id);

ALTER TABLE ai\_interactions ENABLE ROW LEVEL SECURITY;

**Opmerkingen:**

* Dit is een startpunt, specifieke kolommen en types kunnen evolueren.
* Het gebruik van JSONB voor attributes en context biedt flexibiliteit.
* Encryptie van tokens gebeurt in de applicatielaag.
* RLS policies zijn cruciaal en moeten zorgvuldig worden gedefinieerd.
* Indexen zijn essentieel voor performance, vooral op snapshot\_id en api\_name.

**2. Neo4j Graph Model**

Dit model is geoptimaliseerd voor dependency tracing. Data wordt per snapshot opgeslagen.

**2.1 Node Labels en Properties (Herhaling/Verfijning):**

* **:MetadataComponent** (Abstract)
  + snapshotId: STRING (Verplicht, deel van unique constraint)
  + apiName: STRING (Verplicht, deel van unique constraint)
  + label: STRING
  + type: STRING (bijv. 'CustomObject', 'CustomField', 'Flow')
  + sfId: STRING (Optioneel)
  + createdAt: DATETIME (Timestamp van creatie in Neo4j)
* **:CustomObject** (Erft :MetadataComponent)
  + sharingModel: STRING
* **:CustomField** (Erft :MetadataComponent)
  + dataType: STRING
  + isRequired: BOOLEAN
  + formula: STRING (Optioneel)
* **:Flow** (Erft :MetadataComponent)
  + processType: STRING
  + status: STRING ('Active'/'Inactive')
* ... (Andere specifieke labels zoals gedefinieerd in TAD)

**2.2 Relationship Types (Herhaling/Verfijning):**

* BELONGS\_TO: Structurele relatie (bijv. Veld -> Object).
* REFERENCES: Algemene verwijzing tussen componenten (bijv. Flow -> Veld, Flow -> Object).
* USED\_IN: Meer specifieke verwijzing, idealiter met context properties.
  + (field)-[:USED\_IN {elementType: 'Decision'}]->(flow)
* TRIGGERS: Geeft aan dat een trigger reageert op een object (of vice versa).
* IMPACTS: Kan gebruikt worden als een samenvattende relatie na een diepere analyse.

**2.3 Indexen en Constraints (Cypher):**

Deze commando's moeten worden uitgevoerd op de Neo4j database om performance en datakwaliteit te garanderen.

**// Zorgt ervoor dat elke component uniek is binnen een snapshot**

CREATE CONSTRAINT unique\_metadata\_component\_snapshot\_apiName IF NOT EXISTS

FOR (m:MetadataComponent) REQUIRE (m.snapshotId, m.apiName) IS UNIQUE;

**// Index voor snel opzoeken van alle componenten binnen een snapshot**

CREATE INDEX index\_metadata\_component\_snapshotId IF NOT EXISTS

FOR (m:MetadataComponent) ON (m.snapshotId);

**// Index voor snel opzoeken op apiName (handig maar deels gedekt door constraint)**

**// Kan nuttig zijn als je vaak zoekt op apiName ZONDER snapshotId (maar dat zou minder voorkomen)**

CREATE INDEX index\_metadata\_component\_apiName IF NOT EXISTS

FOR (m:MetadataComponent) ON (m.apiName);

**// Index op type voor snelle filtering (bijv. geef alle Flows in snapshot X)**

CREATE INDEX index\_metadata\_component\_type IF NOT EXISTS

FOR (m:MetadataComponent) ON (m.type);

**// Specifieke indexen kunnen nuttig zijn afhankelijk van query patronen, bijv.:**

**// CREATE INDEX index\_flow\_status IF NOT EXISTS FOR (f:Flow) ON (f.status);**

**// CREATE INDEX index\_customfield\_datatype IF NOT EXISTS FOR (cf:CustomField) ON (cf.dataType);**

**Opmerkingen:**

* Het snapshotId is cruciaal in elke node en constraint/index om data te isoleren.
* De keuze tussen REFERENCES en USED\_IN (en de properties op USED\_IN) hangt af van het detailniveau dat we uit de parsing kunnen halen en nodig hebben voor de analyses.