04 - Data Modellen voor NEXA  
Status: IN PROGRESS - (Supabase Schema/RLS/Seed Lokaal OK, Neo4j Model Defined/Setup TODO)  
Inhoudsopgave:

1. Supabase (PostgreSQL) Datamodel
2. Neo4j Graph Model

1. Supabase (PostgreSQL) Datamodel  
Status: [✓ Lokaal Geïmplementeerd & Geseed, ✓ Basis RLS OK]  
Dit beschrijft het gedetailleerde schema voor de PostgreSQL database beheerd door Supabase. RLS policies zijn geïmplementeerd via migraties.

*(Hieronder volgt de SQL DDL zoals gedefinieerd en toegepast via migraties. De status per tabel is effectief '✓ Aangemaakt'.)*

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

-- Status: [✓ Basis OK (Auth users beheerd door Supabase, relaties hieronder OK)]

-- Organisaties (voor multi-tenancy/teams)

-- Status: [✓ Aangemaakt & Geseed]

CREATE TABLE public.organizations (

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

name TEXT NOT NULL,

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

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

ALTER TABLE public.organizations ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

-- Koppeling gebruikers aan organisaties

-- Status: [✓ Aangemaakt & Geseed (voor testgebruiker)]

CREATE TABLE public.organization\_members (

organization\_id uuid REFERENCES public.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',

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

PRIMARY KEY (organization\_id, user\_id)

);

ALTER TABLE public.organization\_members ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

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

-- Status: [✓ Basis OK]

-- Salesforce Connections table

-- Status: [✓ Aangemaakt & Geseed]

CREATE TABLE public.salesforce\_connections (

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

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

sf\_org\_id TEXT NOT NULL UNIQUE,

sf\_user\_id TEXT,

instance\_url TEXT NOT NULL,

access\_token\_encrypted TEXT, -- Status: [Kolom OK, Encryptie Logica TODO]

refresh\_token\_encrypted TEXT, -- Status: [Kolom OK, Encryptie Logica TODO]

status TEXT NOT NULL DEFAULT 'inactive',

last\_connected\_at TIMESTAMPTZ,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

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

);

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

ALTER TABLE public.salesforce\_connections ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

-- Metadata Snapshots table

-- Status: [✓ Aangemaakt]

CREATE TABLE public.metadata\_snapshots (

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

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

status TEXT NOT NULL DEFAULT 'PENDING',

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 public.metadata\_snapshots(sf\_connection\_id);

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

ALTER TABLE public.metadata\_snapshots ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

-- ### Geparsed Metadata Opslag ###

-- Status: [✓ Basis OK]

-- Parsed Components table

-- Status: [✓ Aangemaakt]

CREATE TABLE public.parsed\_components (

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

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

component\_type TEXT NOT NULL,

api\_name TEXT NOT NULL,

label TEXT,

sf\_id TEXT,

attributes JSONB DEFAULT '{}'::jsonb,

-- raw\_definition TEXT,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL,

UNIQUE (snapshot\_id, api\_name)

);

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

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

ALTER TABLE public.parsed\_components ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

-- Parsed Relationships table

-- Status: [✓ Aangemaakt]

CREATE TABLE public.parsed\_relationships (

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

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

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,

context JSONB,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

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

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

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

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

ALTER TABLE public.parsed\_relationships ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

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

-- Status: [✓ Basis OK]

-- Analysis Results table

-- Status: [✓ Aangemaakt]

CREATE TABLE public.analysis\_results (

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

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

analysis\_type TEXT NOT NULL,

component\_api\_name TEXT,

component\_type TEXT,

severity TEXT NOT NULL DEFAULT 'Info',

description TEXT NOT NULL,

details JSONB,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

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

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

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

ALTER TABLE public.analysis\_results ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

-- AI Interactions table

-- Status: [✓ Aangemaakt]

CREATE TABLE public.ai\_interactions (

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

snapshot\_id uuid REFERENCES public.metadata\_snapshots(id) ON DELETE SET NULL,

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)),

feedback\_comment TEXT,

created\_at TIMESTAMPTZ DEFAULT now() NOT NULL

);

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

ALTER TABLE public.ai\_interactions ENABLE ROW LEVEL SECURITY; -- Status: [✓ Basis RLS Toegepast]

**Opmerkingen:**

* Het seed.sql script vult organizations, organization\_members, en salesforce\_connections met testdata. De andere tabellen blijven leeg na supabase db reset.
* Encryptie logica voor tokens is nog niet geïmplementeerd in de (nog te bouwen) Edge Functions.
* RLS policies zijn toegepast, maar vereisen een geldige gebruikerssessie (JWT) om te werken bij API calls.

**2. Neo4j Graph Model  
Status: [✓ Gedefinieerd, Setup TODO]**Dit model is geoptimaliseerd voor dependency tracing. Data wordt per snapshot opgeslagen. De lokale Neo4j database (via Docker) draait, maar de specifieke constraints en indexen voor dit project zijn nog niet toegepast.

**2.1 Node Labels en Properties:**

* **Status: [✓ Gedefinieerd]***(De definities van de labels en hun properties zijn conceptueel klaar)*
* :MetadataComponent (Abstract)
  + Properties: snapshotId: STRING (Verplicht, index, deel van constraint), apiName: STRING (Verplicht, index, deel van constraint), label: STRING, type: STRING (index), sfId: STRING (Optioneel), createdAt: DATETIME
* :CustomObject (Erft :MetadataComponent)
  + Properties: sharingModel: STRING, ...
* :CustomField (Erft :MetadataComponent)
  + Properties: dataType: STRING, isRequired: BOOLEAN, formula: STRING (Optioneel), ...
* :Flow (Erft :MetadataComponent)
  + Properties: processType: STRING, status: STRING, ...
* ... (Andere specifieke labels zoals gedefinieerd in TAD)

**2.2 Relationship Types:**

* **Status: [✓ Gedefinieerd]***(De types relaties die we verwachten te gebruiken zijn conceptueel klaar)*
* 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):**

* **Status: [✓ Gedefinieerd, nog niet toegepast]***(De benodigde Cypher commando's zijn opgeschreven, maar nog niet uitgevoerd op de lokale Neo4j database)*

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

// Status: [ ] Nog niet uitgevoerd

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

// Status: [ ] Nog niet uitgevoerd

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

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

// Index voor snel opzoeken op apiName

// Status: [ ] Nog niet uitgevoerd

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

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

// Index op type voor snelle filtering

// Status: [ ] Nog niet uitgevoerd

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

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

// --- Optionele, specifiekere indexen (kunnen later worden toegevoegd indien nodig) ---

// 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 hangt af van het detailniveau uit de parsing.