

DB Normalisation bzw. Integration

Schema backend vs public

Sebastian Fröhlich

Annahmen und Schlussfolgerung

Gegenüberstellung der Tabellen (frontend/ backend)

Annahme 1:

1. Companies Table wird zu Transactions, aktuell beziehen wir uns nur auf Transaktionen, ein Modell Companies (Firmen) das eine Beziehung zu Transaktionen hat (hasmany), haben wir nicht!

1. Wir haben Transaktionen, die eine Beziehung zum KI Output haben (has many vice versa has one)

2. Somit wird die führende Tabelle companies abgelöst durch die transactions Tabelle

3. Führende Felder die in Companies existieren, aber nicht in transaction hinterlegt sind (rot markiert links)

1. Insbesondere kyc_risk_level benötigen wir für die Ampel (aktuelle Werte: medium, high, low)

4. Felder die in Companies existieren, aber in transactions bzw. transactions output gehören (gelb markiert links)

Annahme 2:

1. Transaction (public) Table wird zu transaction (backend)

1. Alle rot markierten Spalten (links) fehlen und müssen geklärt werden wo diese zu finden sind. Beispiele auf der nächsten Seite

Annahme 3:

1. Die transaction_output bleibt die Tabelle, die nach einen erfolgreichen RUN (Status in backend.transactions) den Output liefert

1. Alle fehlenden Felder (rot markiert) aber insbesondere kyc_risk_level, risk_score/ requires_review sind als Datenfelder verfügbar

2. Das Feld content (rot markiert rechts) ist in irgend einer Form durchsuchbar (json)

table_schema	table_name	column_name	column_default	is_nullable	data_type
public	companies	id	Autoincrement	NO	bigint
public	companies	name	NULL	NO	character varying
public	companies	legal_name	NULL	YES	character varying
public	companies	ticker	NULL	YES	character varying
public	companies	sector	NULL	YES	character varying
public	companies	country	'DE'::character varying	NO	character varying
public	companies	headquarters	NULL	YES	character varying
public	companies	kyc_risk_level	'medium'::character varying	NO	character varying
public	companies	summary	NULL	YES	text
public	companies	created_at	NULL	YES	timestamp without time zone
public	companies	updated_at	NULL	YES	timestamp without time zone

table_schema	table_name	column_name	column_default	is_nullable	data_type
backend	transactions	id	Autoincrement	NO	integer
backend	transactions	corporate_entity	NULL	NO	text
backend	transactions	corporate_counterparty	NULL	NO	text
backend	transactions	tx_date	NULL	NO	text
backend	transactions	tx_amount	NULL	NO	double precision
backend	transactions	tx_currency	NULL	YES	text
backend	transactions	tx_purpose	NULL	YES	text
backend	transactions	tx_country_outgoing	NULL	YES	text
backend	transactions	tx_country_incoming	NULL	YES	text
backend	transactions	source_file	NULL	YES	text
backend	transactions	raw_payload	NULL	YES	text
backend	transactions	status	NULL	NO	text
backend	transactions	created_at	NULL	NO	text
backend	transactions	last_modified_at	NULL	NO	text

table_schema	table_name	column_name	column_default	is_nullable	data_type
public	transactions	id	Autoincrement	NO	bigint
public	transactions	company_id	1	NO	bigint
public	transactions	reference	NULL	NO	character varying
public	transactions	amount	NULL	NO	numeric
public	transactions	currency	'EUR'::character varying	NO	character varying
public	transactions	counterparty	NULL	NO	character varying
public	transactions	counterparty_country	NULL	YES	character varying
public	transactions	channel	NULL	YES	character varying
public	transactions	executed_at	NULL	NO	timestamp without time zone
public	transactions	risk_score	'0'::smallint	NO	smallint
public	transactions	status	NULL	NO	character varying
public	transactions	requires_review	true	NO	boolean
public	transactions	flagged_by	NULL	YES	character varying
public	transactions	flagged_reason	NULL	YES	text
public	transactions	signals	NULL	YES	json
public	transactions	created_at	NULL	YES	timestamp without time zone
public	transactions	updated_at	NULL	YES	timestamp without time zone

table_schema	table_name	column_name	column_default	is_nullable	data_type
backend	transaction_outputs	transaction_id	NULL	NO	integer
backend	transaction_outputs	prompt_id	NULL	NO	integer
backend	transaction_outputs	output_key	NULL	NO	text
backend	transaction_outputs	content	NULL	NO	text
backend	transaction_outputs	run_id	NULL	YES	integer

Normalisation Datenpool

Kopieren der Daten ins Schema public

table_schema	table_name	column_name	Values
backend	transactions	id	1
backend	transactions	corporate_entity	"Mercedes-Benz Group AG"
backend	transactions	corporate_counterparty	"BASF SE"
backend	transactions	tx_date	"2025-10-28"
backend	transactions	tx_amount	54880.9
backend	transactions	tx_currency	"EUR"
backend	transactions	tx_purpose	"Coatings and cathode materials for ba
backend	transactions	tx_country_outgoing	"DE"
backend	transactions	tx_country_incoming	"DE"
backend	transactions	source_file	"transactions-collection.csv"
backend	transactions	raw_payload	"{"Sending Party": "Mercedes-Benz "
backend	transactions	status	Done
backend	transactions	created_at	"2025-11-12T20:59:12.493494+00:00"
backend	transactions	last_modified_at	"2025-11-12T21:01:02.334375+00:00"

table_schema	table_name	column_name	column_default	is_nullable	data_type
backend	transaction_outputs	transaction_id	NULL	NO	integer
backend	transaction_outputs	prompt_id	NULL	NO	integer
backend	transaction_outputs	output_key	NULL	NO	text
backend	transaction_outputs	content	NULL	NO	text
backend	transaction_outputs	run_id	NULL	YES	integer

Annahme 1:

- Wir schaffen ein Datenpool, in einer neuen Tabelle, die ein left join aus transactions und transaction_outputs beinhaltet, diese neue Tabelle ist die Grundlage für jede weitere Datennormalisierung und muss regelmäßig im Hintergrund das backend Schema abfragen:

- Der SQL Syntax lautet in etwa so:

```
Select * from backend.transactions as t
left join backend.transaction_outputs as tout on t.id = tout.transaction_id
where t.status = 'done'
order by transaction_id, prompt_id;
```

- Für eine Normalisation der Datenbank hat das public schema ein neues Datenbank Feld: content_json (rot markiert rechts). Dies wird für die weitere Normalisation benötigt in die Tabellen Companies und transactions im public Schema

- Der SQL Syntax lautet in etwa so:

```
UPDATE public.transaction_outputs
SET content_json = content::jsonb
WHERE content IS NOT NULL;
```

Der Schritt 1 lässt sich als einfacher „Kopierbefehl“ absetzen, wobei man genau die Spalten im Select definieren muss, da durch die neue content_json Spalte die Spaltenanzahl sich sonst nicht gleicht und der Befehl nicht funktionieren würde (Das Beispiel muss entsprechend Angepasst werden:

Beispiel:

```
/* Neue werte einfügen, die noch nicht kopiert worden sind */
INSERT INTO public.transaction_outputs
SELECT * FROM backend.transaction_outputs
EXCEPT
SELECT * FROM public.transaction_outputs;
```

table_schema	table_name	column_name	Values
public	transaction_outputs	id	1
public	transaction_outputs	corporate_entity	"Mercedes-Benz Group AG"
public	transaction_outputs	corporate_counterparty	"BASF SE"
public	transaction_outputs	tx_date	"2025-10-28"
public	transaction_outputs	tx_amount	54880.9
public	transaction_outputs	tx_currency	"EUR"
public	transaction_outputs	tx_purpose	"Coatings and cathode materials for ba
public	transaction_outputs	tx_country_outgoing	"DE"
public	transaction_outputs	tx_country_incoming	"DE"
public	transaction_outputs	source_file	"transactions-collection.csv"
public	transaction_outputs	raw_payload	"{"Sending Party": "Mercedes-Benz "
public	transaction_outputs	status	Done
public	transaction_outputs	created_at	"2025-11-12T20:59:12.493494+00:00"
public	transaction_outputs	last_modified_at	"2025-11-12T21:01:02.334375+00:00"
public	transaction_outputs	transaction_id	1
public	transaction_outputs	prompt_id	1
public	transaction_outputs	output_key	"corporate_summary"
public	transaction_outputs	content	"{"answer": "BASF SE is a German..
public	transaction_outputs	run_id	102
public	transaction_outputs	content_json	"{"answer": "BASF SE is a German..

Normalisation Datenpool

Erweiterung der transactions table im Public Schema

Annahme 1:

- Die Tabelle transactions muss um die Felder erweitert werden, die im neuen Datenpool im Feld output_key ausgegeben werden (quasi unsere variablenamen) als typ-> jsonb

public_transactions_scheme					
table_schema	table_name	column_name	column_default	is_nullable	data_type
public	transactions	id	Autoincrement	NO	bigint
public	transactions	company_id	1	NO	bigint
public	transactions	reference	NULL	NO	character varying
public	transactions	amount	NULL	NO	numeric
public	transactions	currency	'EUR'::character varying	NO	character varying
public	transactions	counterparty	NULL	NO	character varying
public	transactions	counterparty_country	NULL	YES	character varying
public	transactions	channel	NULL	YES	character varying
public	transactions	executed_at	NULL	NO	timestamp without time zone
public	transactions	risk_score	'0'::smallint	NO	smallint
public	transactions	status	NULL	NO	character varying
public	transactions	requires_review	true	NO	boolean
public	transactions	flagged_by	NULL	YES	character varying
public	transactions	flagged_reason	NULL	YES	text
public	transactions	signals	NULL	YES	json
public	transactions	created_at	NULL	YES	timestamp without time zone
public	transactions	updated_at	NULL	YES	timestamp without time zone

Annahme 2:

- Einige Felder müssen danach weiter verarbeitet werden und in die Companies Table gemappt werden.

backend_transaction_outputs

output_key
corporate_summary
corporate_history
corporate_purpose
corporate_sector
corporate_nace
corporate_products
corporate_markets
corporate_supply
corporate_name
corporate_form
corporate_forum
corporate_HQ
corporate_locations
corporate_holding
corporate_shareholders
corporate_board
corporate_supervisory
corporate_powerofattorney
corporate_taxID
corporate_LEI
corporate_UBO
corporate_employeecount
corporate_turnover
corporate_EBIT
corporate_netprofits
corporate_balancesheet
corporate_auditfindings
corporate_insolvency
corporate_liquidation
corporate_adhoc
corporate_pressrelease
corporate_votes
corporate_directordealings
corporate_directordealings

backend_transaction_outputs

output_key
corporate_brands
corporate_website
corporate_domain
corporate_IBAN
corporate_solvency
corporate_rating
corporate_ESG
corporate_license
corporate_warnings
corporate_eusanctions
corporate_ofacsanctions
corporate_uksanctions
corporate_pepexposure
corporate_adversemediascanning
corporate_corruptionexposure
corporate_AMLexposure
corporate_CTyexposure
tranx_TX_AMOUNT
tranx_historical
tranx_TX_PURPOSE
tranx_counterpartyassessment
tranx_context
tranx_report
tranx_PURPOSEbase
tranx_performanceperiod
tranx_seasonality
tranx_resolution
trans_IBANvalidation
tranx_businesslogic
source_domaincheck
tranx_plausibility
tranx_outliers
tranx_patterns
tranx_revenueimpact

backend_transaction_outputs

output_key
tranx_profitimpact
tranx_marketimpact
source_check
source_approveuniqueID
sanctions_circumvention
corruption_sector
corruption_country
corruption_relationship
corporate_adverse
corporate_pep
corporate_adverse2
tranx_duplicate
tranx_pattern2
corporate_exportcontrol
corporate_mediamatch
tranx_contracttenor
corporate_haven
tranx_mismatch
country_risk
corporate_haven2
corporate_insiders
corporate_courtcases
corporate_manda
tranx_fakepurposecheck
tranx_structuring
tranx_newbankaccount
corporate_pepdetail
tranx_weekday
tranx_holdingobfuscation
source_randomplausibility
tranx_score
tranx_reasoning
tranx_recommendation
tranx_sourcerepository

Normalisation Datenpool

Mapping aus dem Datenpool in die companies Table

public_companies_scheme			
table_schema	table_name	column_name	Values
public	companies	id	1
public	companies	name	"Adidas AG"
public	companies	legal_name	"adidas Aktiengesellschaft"
public	companies	ticker	"ADS"
public	companies	sector	"Konsumgüter (Zyklisch)"
public	companies	country	"DE"
public	companies	headquarters	"Herzogenaurach"
public	companies	kyc_risk_level	"medium"
public	companies	summary	"Globaler Sportartikelhersteller mit Sch
public	companies	created_at	"2025-10-21 11:52:36"
public	companies	updated_at	"2025-10-21 11:52:36"

Annahme 1:

- Die gelb markierten Felder (Achtung Datenqualität) kann aus der public transactions_output gemappt werden bzw. damit gefüllt werden
- Das pink markierte Feld (ticker) ist nicht klar, woher es kommt , handelt es sich bei einem börsennotierten Unternehmen um das symbol/ Ticker? - **Klärung!**
- Die grün markierten Felder können ebenfalls aus dem Datenpool übernommen werden, sie kommen jedoch auch dem „backend“ Schema transactions und sind direkte Textfelder, die daher auch nicht über aus einem Json Feld kommen

Datenqualität:

- 96 Datensätze erfolgreich bereinigt (valides JSON)
- 43 Datensätze mit Syntaxfehlern (können nicht geparst werden)
- 7.635 Text-Antworten (unverändert)

Empfehlung:

Die Datenqualität ist nicht konstant -

96 Datenfelder haben im answer-Feld doppelte JSON-encoded (String mit Escape-Zeichen \")

3 JSON-Strings haben Syntaxfehler in den sources - es fehlen schließende Klammern }

Die 43 fehlerhaften sollten wir als Strings belassen. Das Problem liegt in der Datenquelle - vermutlich ein Bug beim Generieren der Antworten.

public_transaction_outputs_o

output_key
corporate_summary
corporate_history
corporate_purpose
corporate_sector
corporate_nace
corporate_products
corporate_markets
corporate_supply
corporate_name
corporate_form
corporate_forum
corporate_HQ
corporate_locations
corporate_holding
corporate_shareholders
corporate_board
corporate_supervisory
corporate_powerofattorney
corporate_taxID
corporate_LEI
corporate_UBO
corporate_employeecount
corporate_turnover
corporate_EBIT
corporate_netprofits
corporate_balancesheet
corporate_auditfindings
corporate_insolvency
corporate_liquidation
corporate_adhoc
corporate_pressrelease
corporate_votes
corporate_directordealings
corporate_directordealings

public_transaction_outputs_o

output_key
corporate_brands
corporate_website
corporate_domain
corporate_IBAN
corporate_solvency
corporate_rating
corporate_ESG
corporate_license
corporate_warnings
corporate_eusanctions
corporate_ofacsanctions
corporate_uksanctions
corporate_pepexposure
corporate_adversemediascanning
corporate_corruptionexposure
corporate_AMLexposure
corporate_CTYexposure
tranx_TX_AMOUNT
tranx_historical
tranx_TX_PURPOSE
tranx_counterpartyassessment
tranx_context
tranx_report
tranx_PURPOSEbase
tranx_performanceperiod
tranx_seasonality
tranx_resolution
trans_IBANvalidation
tranx_businesslogic
source_domaincheck
tranx_plausibility
tranx_outliers
tranx_patterns
tranx_revenueimpact

public_transaction_outputs_o

output_key
tranx_profitimpact
tranx_marketimpact
source_check
source_approveuniqueID
sanctions_circumvention
corruption_sector
corruption_country
corruption_relationship
corporate_adverse
corporate_pep
corporate_adverse2
tranx_duplicate
tranx_pattern2
corporate_exportcontrol
corporate_mediamatch
tranx_contracttenor
corporate_haven
tranx_mismatch
country_risk
corporate_haven2
corporate_insiders
corporate_courtcases
corporate_manda
tranx_fakepurposecheck
tranx_structuring
tranx_newbankaccount
corporate_pepdetail
tranx_weekday
tranx_holdingobfuscation
source_randomplausibility
tranx_score
tranx_reasoning
tranx_recommendation
tranx_sourcerepository

Status

16.11.2025

New Branch feature/integration-frontend-backend

Migration File & Job Scheduler für inkremtiellen- und Full-Sync wurden eingebaut

102 Spalten wurden als jsonb Felder zur transactions Tabelle hinzugefügt: Kategorien:

- Corporate Information (61 Spalten): Firmeninformationen wie Summary, History, Shareholder, UBO, Finanzdaten, Sanktionen, etc.
- Transaction Information (33 Spalten): Transaktionsdaten wie Amount, Purpose, Historical, Scoring, Recommendations, etc.
- Source Information (4 Spalten): Domain-Checks, Approval IDs, Plausibility, etc.
- Risk Assessment (5 Spalten): Sanctions circumvention, Corruption exposure, Country risk, etc.

Laravel Scheduler (Empfohlen für Produktion)

Der Laravel Scheduler benötigt einen Cron Job, der jede Minute läuft.

1. 1. Cron Job einrichten

1. `***** cd /<PATH> && php artisan schedule:run >> /dev/null 2>&1`

2. Oder für besseres Logging:

1. `***** cd <PATH> && php artisan schedule:run >> <PATH FRONTEND>/storage/logs/scheduler.log 2>&1`

3. Schedule List prüfen

1. `php artisan schedule:list`

```
> php artisan schedule:list
0 */6 * * * Sync new/updated backend transactions to data pool ..... Next Due: 3 hours from now
0 3 * * 0 Full sync and rebuild of backend data pool ..... Next Due: 5 days from now
```

**WICHTIG - Festlegung, dass sich jetzt am DB
SCHEMA von backend nichts mehr ändert!!!**

Annahme 1:

- 1. Companies Table wird zu Transactions, aktuell beziehen wir uns nur auf Transaktionen, ein Modell Companies (Firmen) das eine Beziehung zu Transaktionen hat (hasmany), haben wir nicht!

- 1. Wir haben Transaktionen, die eine Beziehung zum KI Output haben (has many vice versa has one)

- 2. Somit wird die führende Tabelle companies abgelöst durch die transactions Tabelle

- 3. Führende Felder die in Companies existieren, aber nicht in transaction hinterlegt sind (rot markiert links)

- 1. Insbesondere kyc_risk_level benötigen wir für die Ampel (aktuelle Werte: medium, high, low)

- 4. Felder die in Companies existieren, aber in transactions bzw. transactions output gehören (gelb markiert links)

public_companies_scheme			
table_schema	table_name	column_name	Values
public	companies	id	1
public	companies	name	"Adidas AG"
public	companies	legal_name	"adidas Aktiengesellschaft"
public	companies	ticker	"ADS"
public	companies	sector	"Konsumgüter (Zyklisch)"
public	companies	country	"DE"
public	companies	headquarters	"Herzogenaurach"
public	companies	kyc_risk_level	"medium"
public	companies	summary	"Globaler Sportartikelhersteller mit Sch
public	companies	created_at	"2025-10-21 11:52:36"
public	companies	updated_at	"2025-10-21 11:52:36"

backend_transactions_scheme			
table_schema	table_name	column_name	Values
backend	transactions	id	1
backend	transactions	corporate_entity	"Mercedes-Benz Group AG"
backend	transactions	corporate_counterparty	"BASF SE"
backend	transactions	tx_date	"2025-10-28"
backend	transactions	tx_amount	54880.9
backend	transactions	tx_currency	"EUR"
backend	transactions	tx_purpose	"Coatings and cathode materials for ba
backend	transactions	tx_country_outgoing	"DE"
backend	transactions	tx_country_incoming	"DE"
backend	transactions	source_file	"transactions-collection.csv"
backend	transactions	raw_payload	"{"Sending Party": ""Mercedes-Benz "
backend	transactions	status	Done
backend	transactions	created_at	"2025-11-12T20:59:12.493494+00:00"
backend	transactions	last_modified_at	"2025-11-12T21:01:02.334375+00:00"

public_transactions_scheme					
table_schema	table_name	column_name	column_default	is_nullable	data_type
public	transactions	id	Autoincrement	NO	bigint
public	transactions	company_id		NO	bigint
public	transactions	reference	NULL	NO	character varying
public	transactions	amount	NULL	NO	numeric
public	transactions	currency	'EUR'::character varying	NO	character varying
public	transactions	counterparty	NULL	NO	character varying
public	transactions	counterparty_country	NULL	YES	character varying
public	transactions	channel	NULL	YES	character varying
public	transactions	executed_at	NULL	NO	timestamp without time zone
public	transactions	risk_score	'0'::smallint	NO	smallint
public	transactions	status	NULL	NO	character varying
public	transactions	requires_review	true	NO	boolean
public	transactions	flagged_by	NULL	YES	character varying
public	transactions	flagged_reason	NULL	YES	character varying
public	transactions	signals	NULL	YES	character varying
public	transactions	created_at	NULL	YES	timestamp without time zone
public	transactions	updated_at	NULL	YES	timestamp without time zone

public_transaction_outputs_scheme					
table_schema	table_name	column_name	column_default	is_nullable	data_type
public	transaction_outputs	transaction_id	NULL	NO	integer
public	transaction_outputs	prompt_id	NULL	NO	integer
public	transaction_outputs	output_key	NULL	NO	text
public	transaction_outputs	content	NULL	NO	text
public	transaction_outputs	run_id	NULL	YES	integer
public	transaction_outputs	content_json	NULL	YES	jsonb