

Pilot Moodle-DWH

Ein Pilot-DWH in der Oracle Cloud

Technische Umsetzung vom Bucket bis zum Dashboard

Dr. Andrea Kennel, fhnw

22. November 2024



Dr. Andrea Kennel

_SYM^{L2}



Consultant

Dozentin für Datenbanken

Coach für Project Management

Fachhochschule Nordwestschweiz

Brugg/Windisch, Schweiz

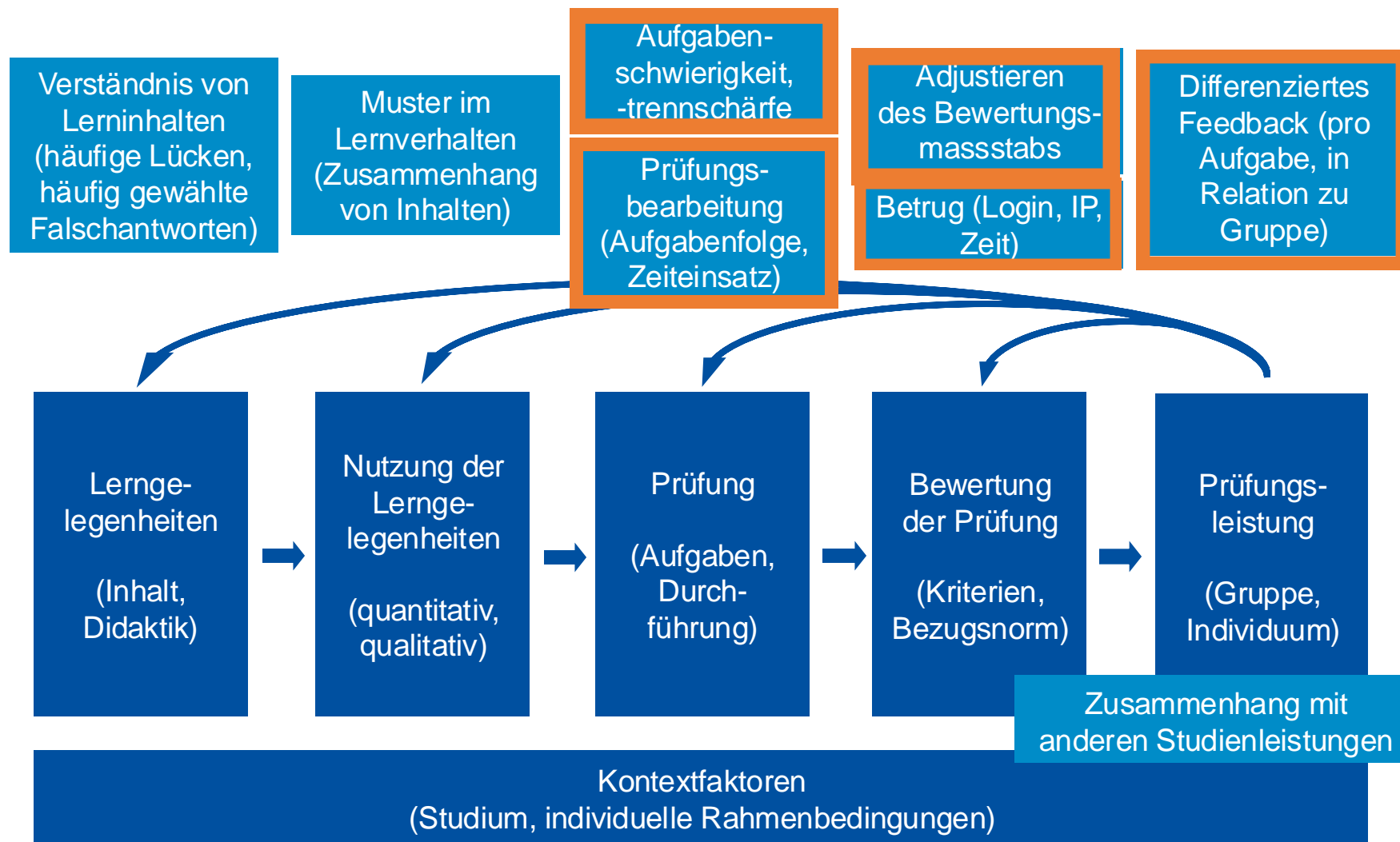


andrea.kennel@fhnw.ch

andrea@infokennel.ch

www.infokennel.ch

Grundlage



Beispiel Zeiteinsatz

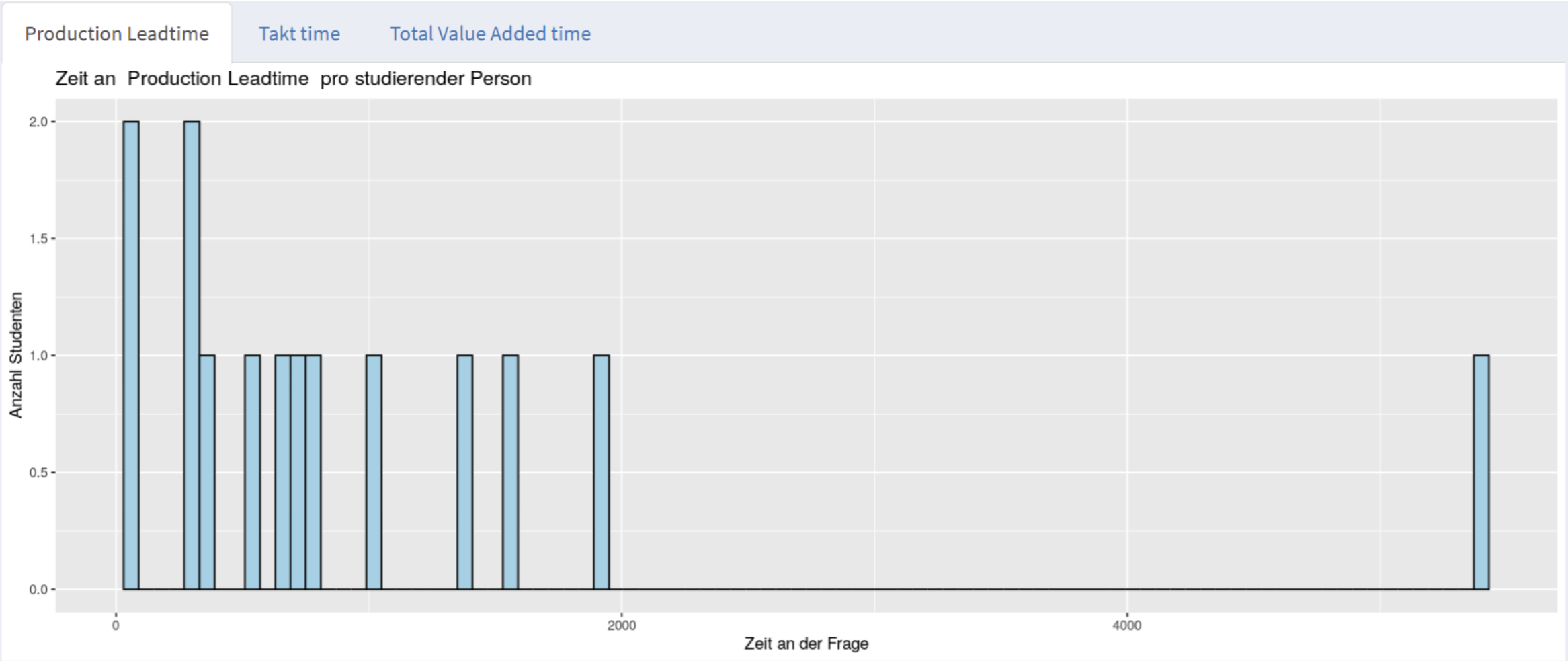
Antworten-Rückblick

Schritt	Zeit	Aktion	Status	Punkte
1	18. April 2024, 13:59:09	Begonnen	Bisher nicht beantwortet	
2	18. April 2024, 14:00:59	Gespeichert: Diese Antwort ist die zweite falsche Antwort und ist mit -40% gewichtet	Antwort gespeichert	
3	18. April 2024, 14:01:37	Gespeichert: Diese Antwort ist die erste falsche Antwort und ist mit -60% gewichtet	Antwort gespeichert	
4	18. April 2024, 14:11:07	Versuch beendet	Falsch	-6,00

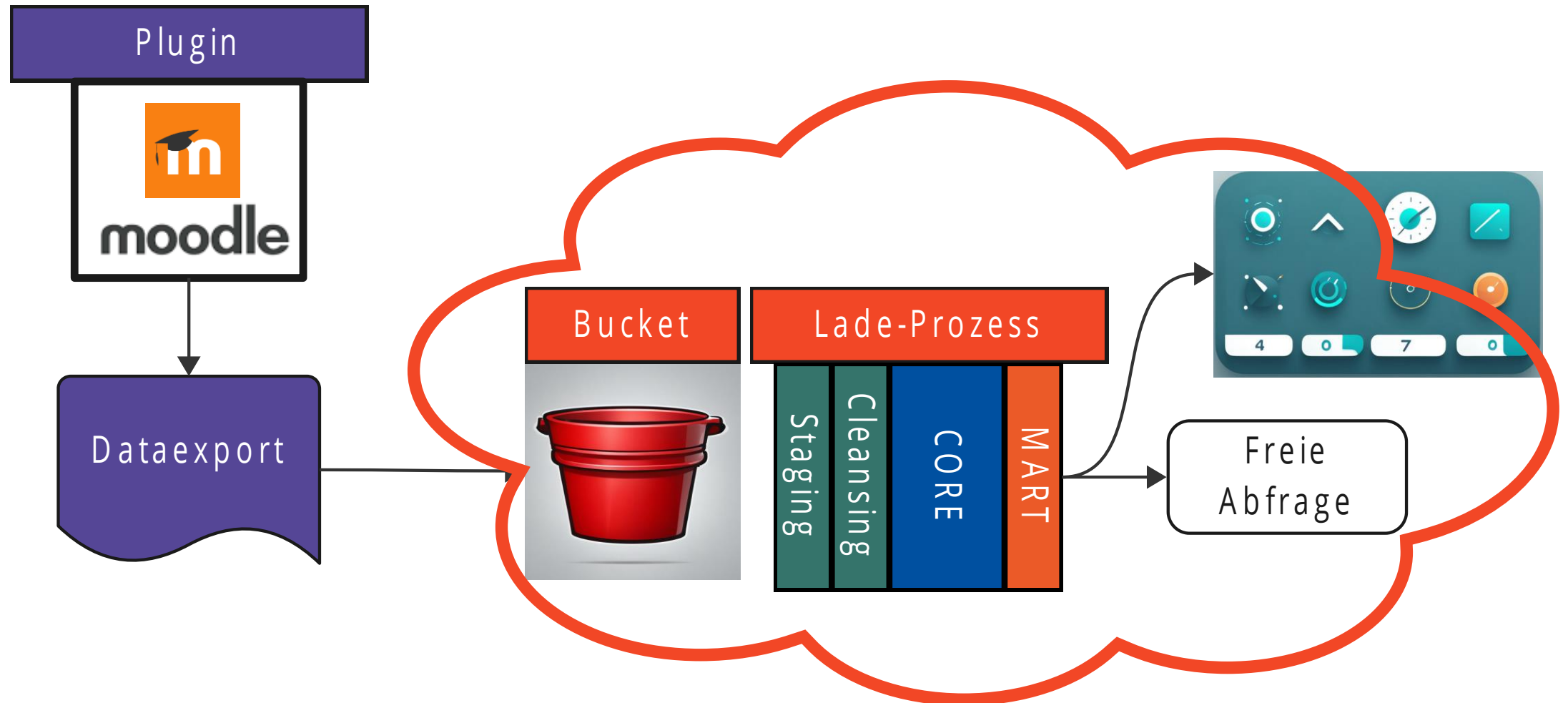
Antworten-Rückblick

Schritt	Zeit	Aktion	Status	Punkte
1	18. April 2024, 13:59:09	Begonnen	Bisher nicht beantwortet	
2	18. April 2024, 14:10:58	Gespeichert: Test Text zum identifizieren	Antwort gespeichert	
3	18. April 2024, 14:11:07	Versuch beendet	Vollständig	

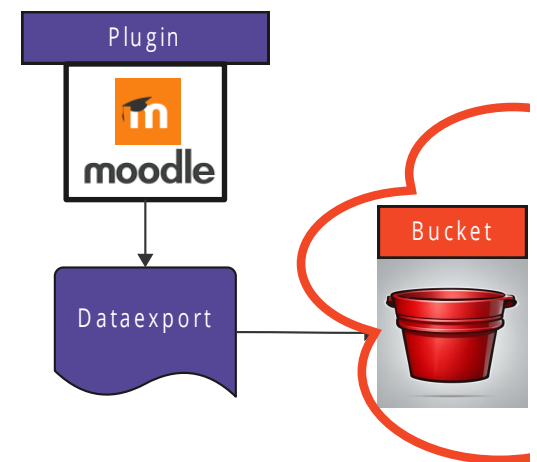
Beispiel Zeiteinsatz



Übersicht



Von Moodle ins Bucket



Resources

- Objects
- Metrics
- Pre-Authenticated Request**
- Work Requests
- Lifecycle Policy Rules
- Replication Policy
- Retention Rules
- Uncommitted Multipart Uploads
- Logs

Create Pre-Authenticated Request

Search by object prefix

Name: par-bucket-20231121-1131

Bucket

Create a pre-authenticated request that applies to all objects in the bucket.

Access Type

- ☐ Permit object read-only
- ☐ Permit object write
- ☒ Permit object read and write

☐ Enable Object Listing

Let users list the objects in the bucket.

Expiration: Nov 28, 2023 10:31 UTC

Create Pre-Authenticated Request [Cancel](#)

Pre-Authenticated Request Details

Name *Read-only*: par-bucket-20231121-1131

Pre-Authenticated Request URL *Read-only*: <https://objectstorage.eu-zurich-1.oraclecloud.com/p/par-bucket-20231121-1131/b/bucket-20231121-0950/>

Copy this URL for your records. It will not be shown again.

The current URL is deprecated and will no longer be supported in a future release of the CLI. [Learn more](#)

<https://zr9o97rpuj34.objectstorage.eu-zurich-1.oci.customer-oci.com/p/par-bucket-20231121-1131/b/bucket-20231121-0950/>

Close

Edit backend

Name: DWH_24

Description: Bucketzugriff bis Dez 24

URL: <https://objectstorage.eu-zurich-1.oraclecloud.com/p/par-bucket-20231121-1131/b/bucket-20231121-0950/>

Username:

Password:

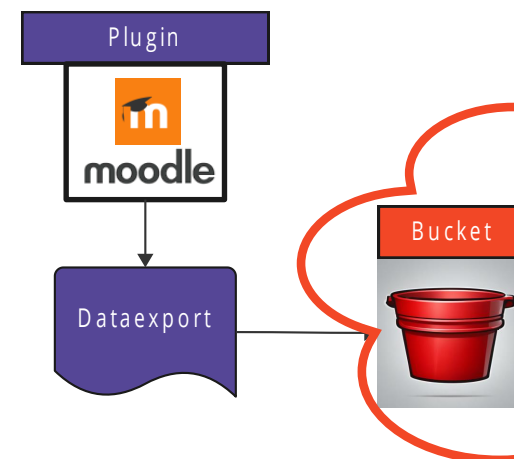
Enabled: Ja

Allowed user:

[Änderungen speichern](#) [Abbrechen](#)

notwendig

Von Moodle ins Bucket mit PHP-Funktion `curl_exec`



```
// PUT to a Pre-Authenticated Requests enabled Oracle Object Storage Bucket.
$url = $DB->get_field('report_datawarehouse_bkends', 'url', ['id' => $backendid]);
// Initiate cURL object.
$curl = curl_init();
// Set your URL.
curl_setopt($curl, CURLOPT_URL, $url . $filename);
// Indicate your protocol.
curl_setopt($curl, CURLOPT_PROTOCOLS, CURLPROTO_HTTPS);
// Set HTTP method to PUT.
curl_setopt($curl, CURLOPT_PUT, 1);
// Indicate the file you want to upload.
curl_setopt($curl, CURLOPT_INFILE, fopen($tempfolder . '/' . $filename, 'rb'));
...
// Execute.
curl_exec($curl);
```


Datenschutz

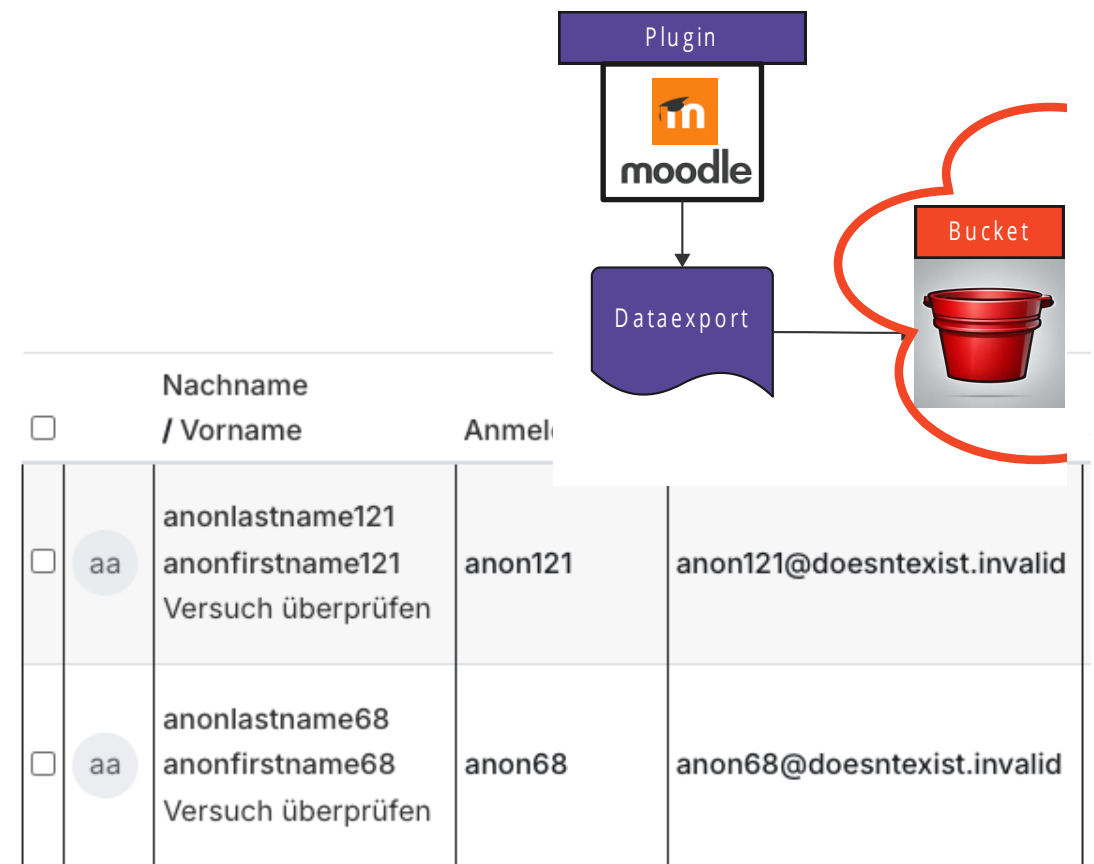
User ist doppelt anonymisiert

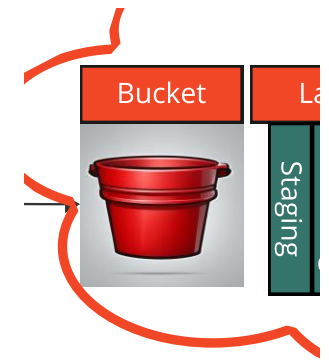
1. In Moodle User anonymisieren

2. Nur Hash exportieren

SELECT

```
...
(SELECT MD5(username) FROM mdl_user WHERE id = qzat.userid) AS "qzatuserid",
...
```





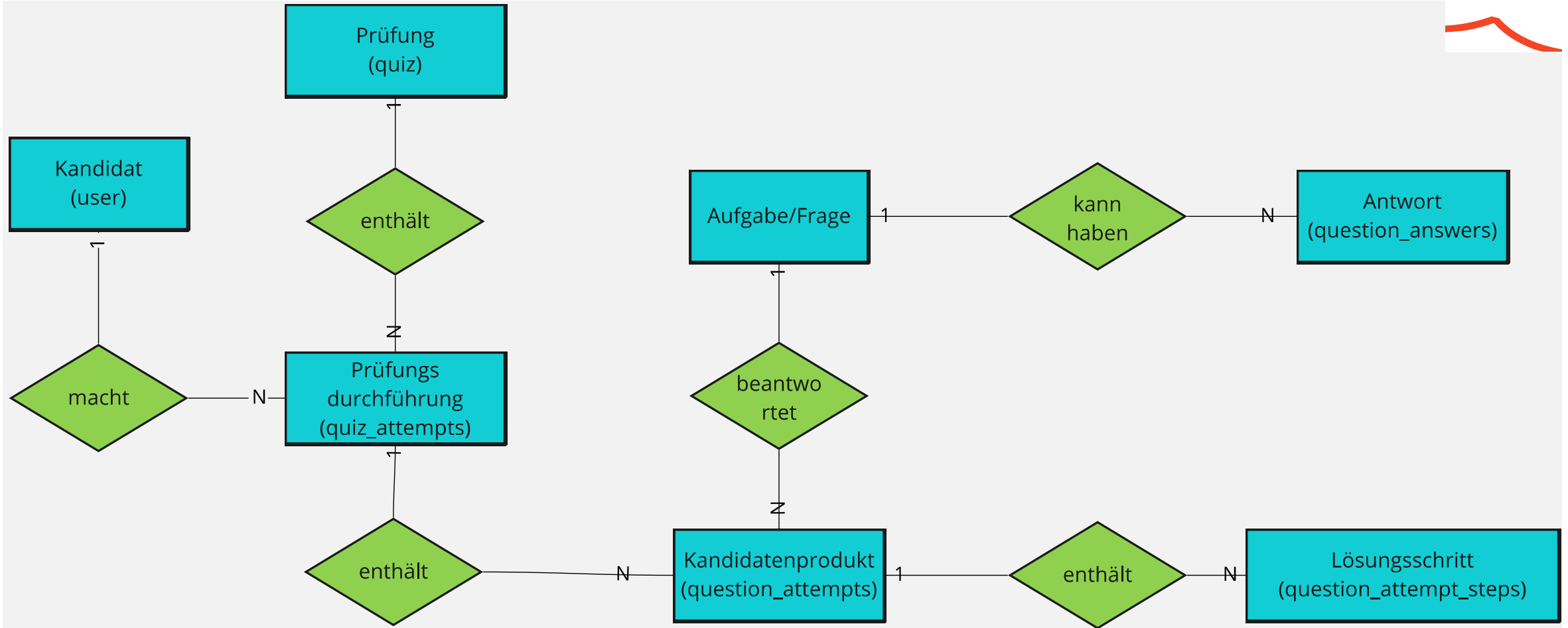
Vom Bucket in die Stage mit DBMS_CLOUD.COPY_DATA

```
"C##CLOUD$SERVICE"."DBMS_CLOUD"."COPY_DATA"
( TABLE_NAME          => l_TABLE_NAME
, CREDENTIAL_NAME      => l_CREDENTIAL_NAME
, FILE_URI_LIST        => l_FILE_URI_LIST
, FIELD_LIST           => l_FIELD_LIST
, FORMAT               => l_FORMAT
, OPERATION_ID         => l_OPERATION_ID
);
```

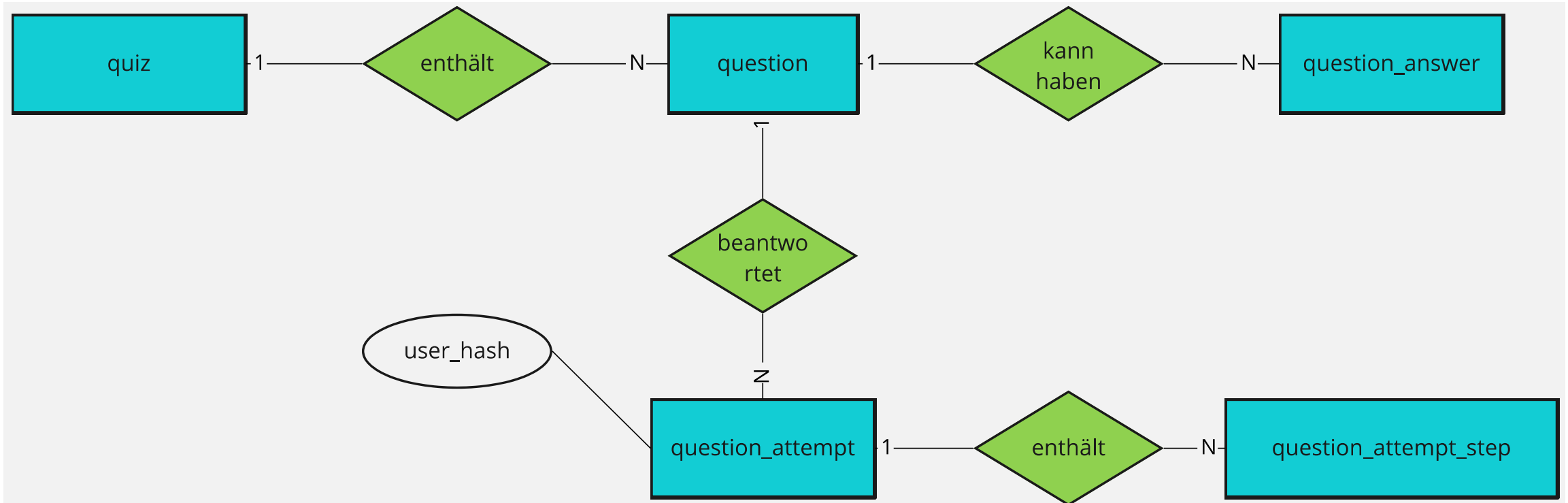
```
v_full_path VARCHAR2(2000) := p_bucket_path || p_file_name;

-- DECLARE
l_TABLE_NAME          DBMS_QUOTED_ID := '"DATA_VERSION_BACH"';
l_CREDENTIAL_NAME     DBMS_QUOTED_ID := '"OBJ_STORE_CRED"';
l_FILE_URI_LIST       CLOB := v_full_path;
l_FIELD_LIST          CLOB :=
q' [
    "SOURCESYSTEM"          CHAR(4000)
    , "AUTHORIZEDUSERS"     CHAR(4000)
    , "QUESTIONID"          CHAR
```

Das Datenmodell der Quelle



Das Modell in der Cleansing

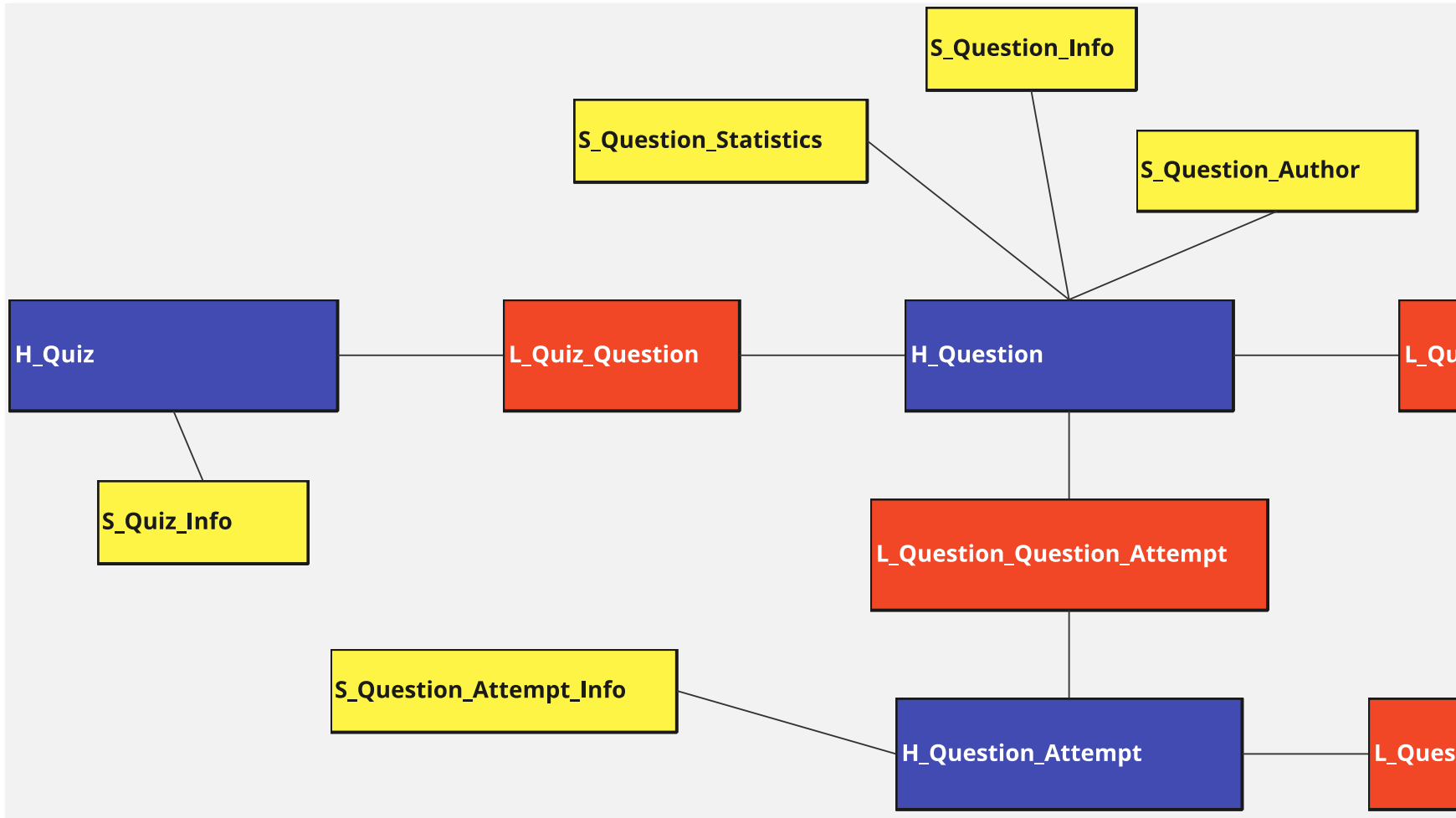


Von der Stage in die Cleansing

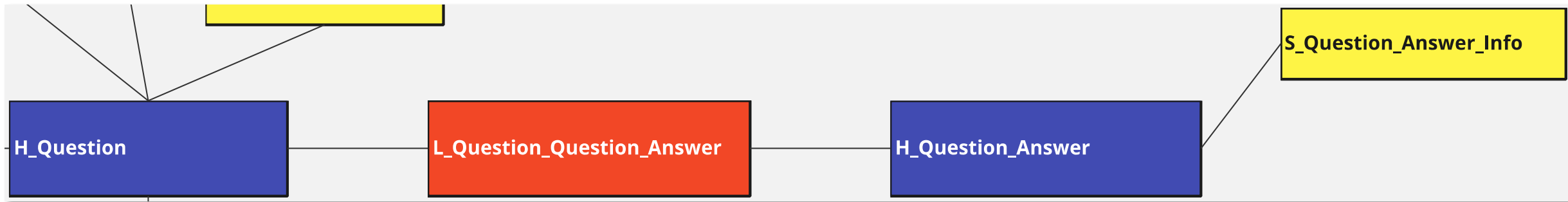
```
INSERT INTO MD_CLEANSING.quiz (  
    load_id,  
    sourcesystem,  
    quiz_id,  
    authorizedusers,  
    quiz_name,  
    quiz_grade)  
SELECT DISTINCT  
    p_load_id load_id,  
    sourcesystem,  
    quizid quiz_id,  
    authorizedusers,  
    quizname quiz_name,  
    quizgrade quiz_grade  
FROM MD_STAGING.data_view d;
```

```
INSERT INTO MD_CLEANSING.question_attempt_step_data (  
    load_id,  
    sourcesystem,  
    authorizedusers,  
    question_attempt_step_data_id,  
    question_attempt_step_id,  
    step_data_name,  
    step_data_value)  
SELECT DISTINCT  
    p_load_id load_id,  
    sourcesystem sourcesystem,  
    authorizedusers authorizedusers,  
    questionattemptstepdataid question_attempt_step_data_id,  
    qastepid question_attempt_step_id,  
    questionattemptstepdataname step_data_name,  
    questionattemptstepdatavalue step_data_value  
FROM MD_STAGING.data_view  
WHERE questionattemptstepdataid IS NOT NULL;
```

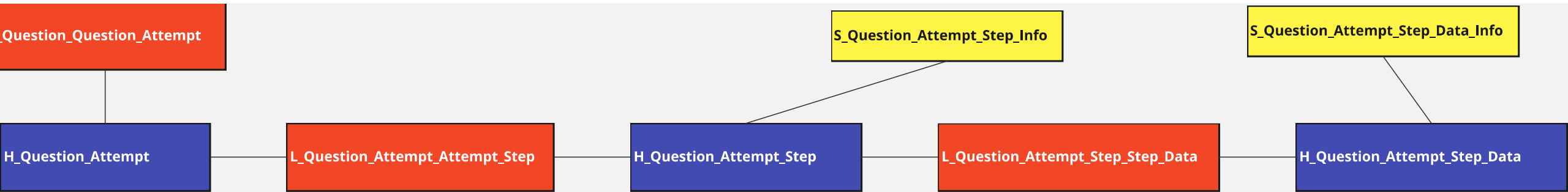
Das Modell im Core



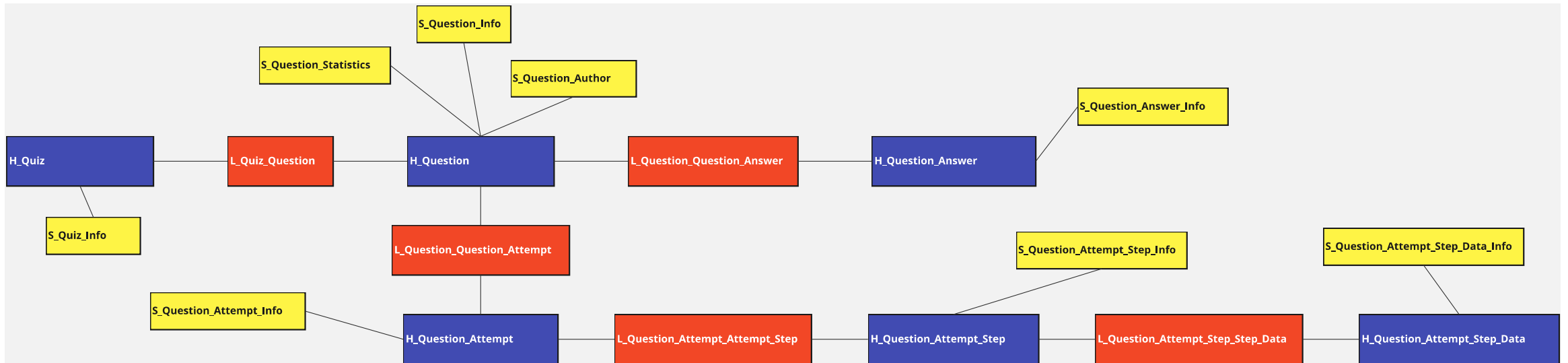
Das Modell im Core



Das Modell im Core



Das Modell im Core



Von der Cleansing ins Core

1. HUB füllen

Künstlicher Schlüssel für HUB

H_Question

h_question_sid

Quellschlüssel

question_id
load_id

Von der Cleansing ins Core

2. Satelliten zu HUB füllen

S_Question_info

Fremdschlüssel auf HUB via
Quellschlüssel finden

h_question_sid

```
SELECT
    h.h_question_sid,
    cl.question_type, cl.question_name,
    cl.question_text
FROM MD_CLEANSING.question cl INNER JOIN
    H_Question h ON (cl.question_id = h.question_id
                     AND cl.load_id = h.load_id);
```

Von der Cleansing ins Core

3. Link zwischen HUB füllen

Künstlicher Schlüssel für Link

Beide Fremdschlüssel auf
HUB via Quellschlüssel finden

L_Question_Question_Answer

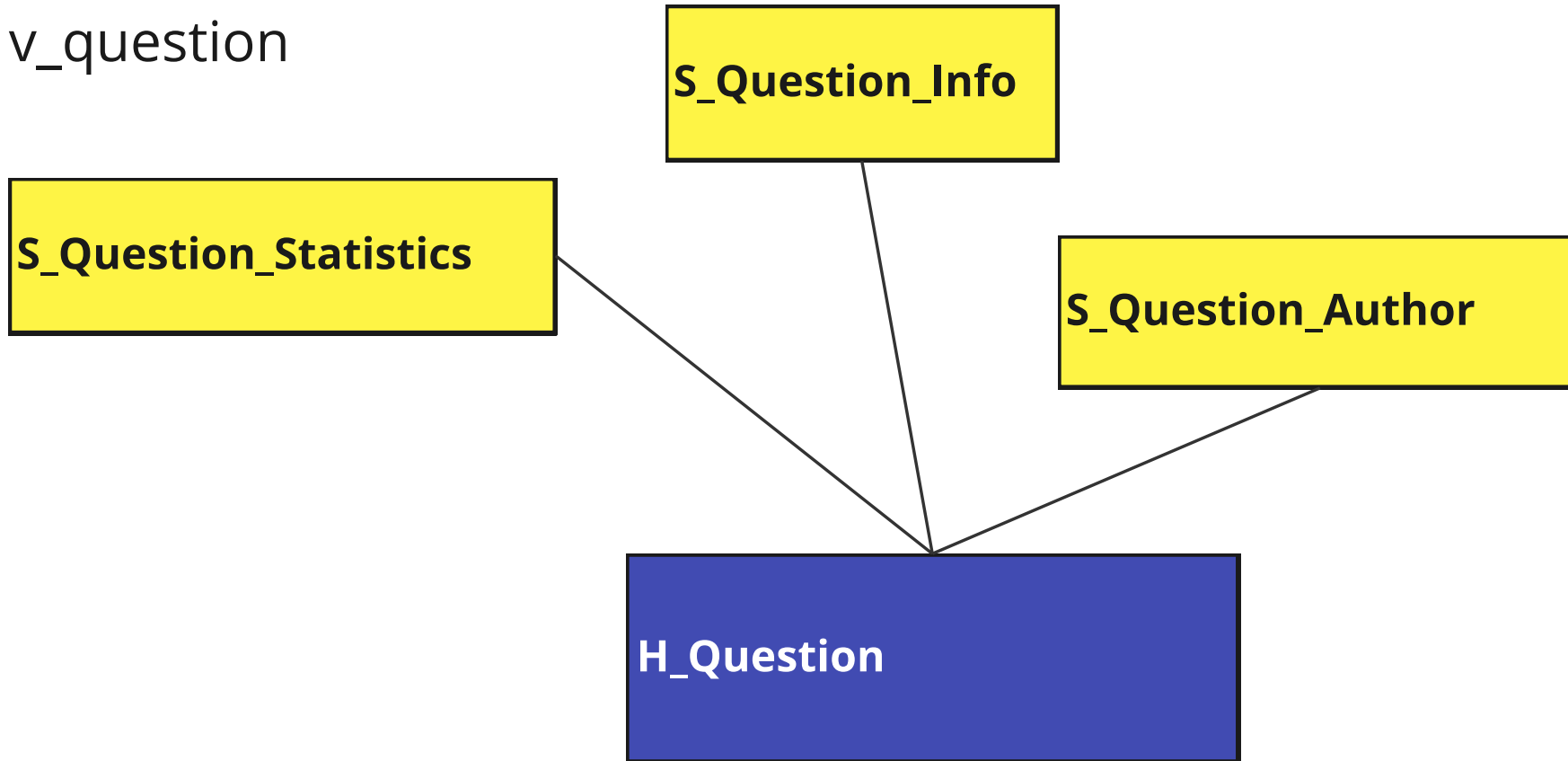
l_question_question_answer_sid

h_question_sid

h_question_answer_sid

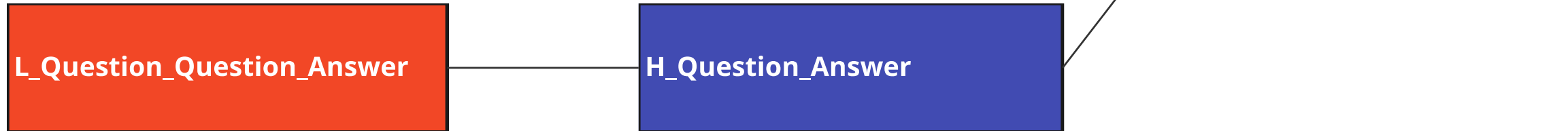
Views für einfacheren Zugriff auf Core

v_question



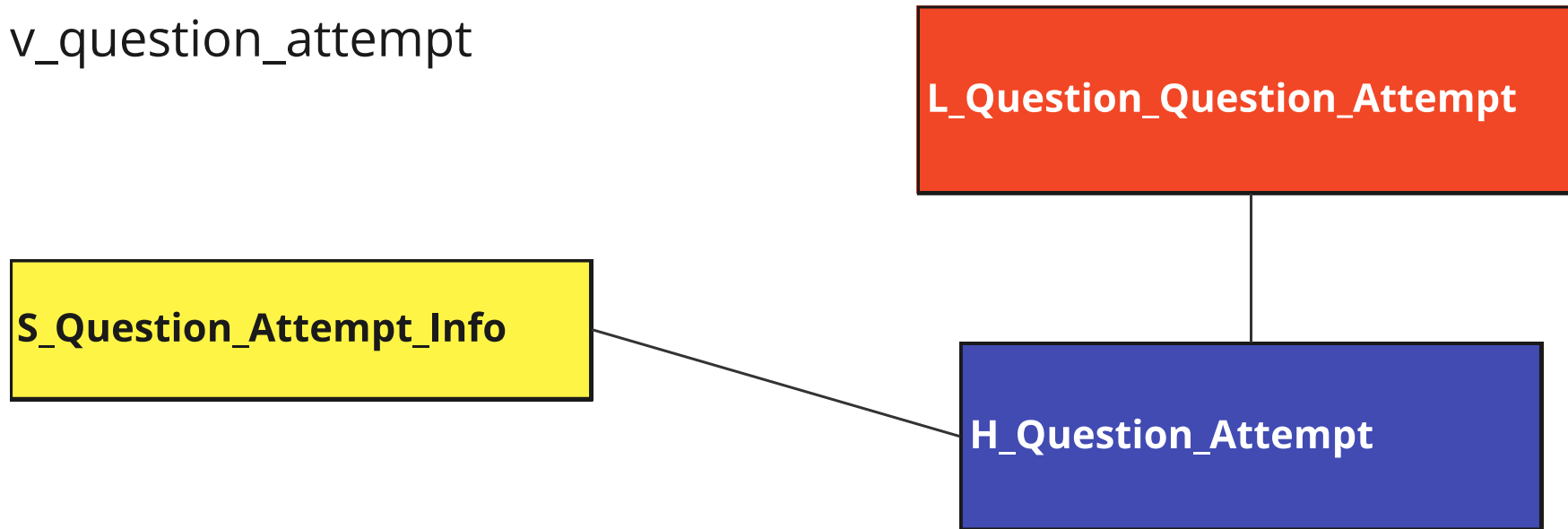
Views für einfacheren Zugriff auf Core

v_question_answer

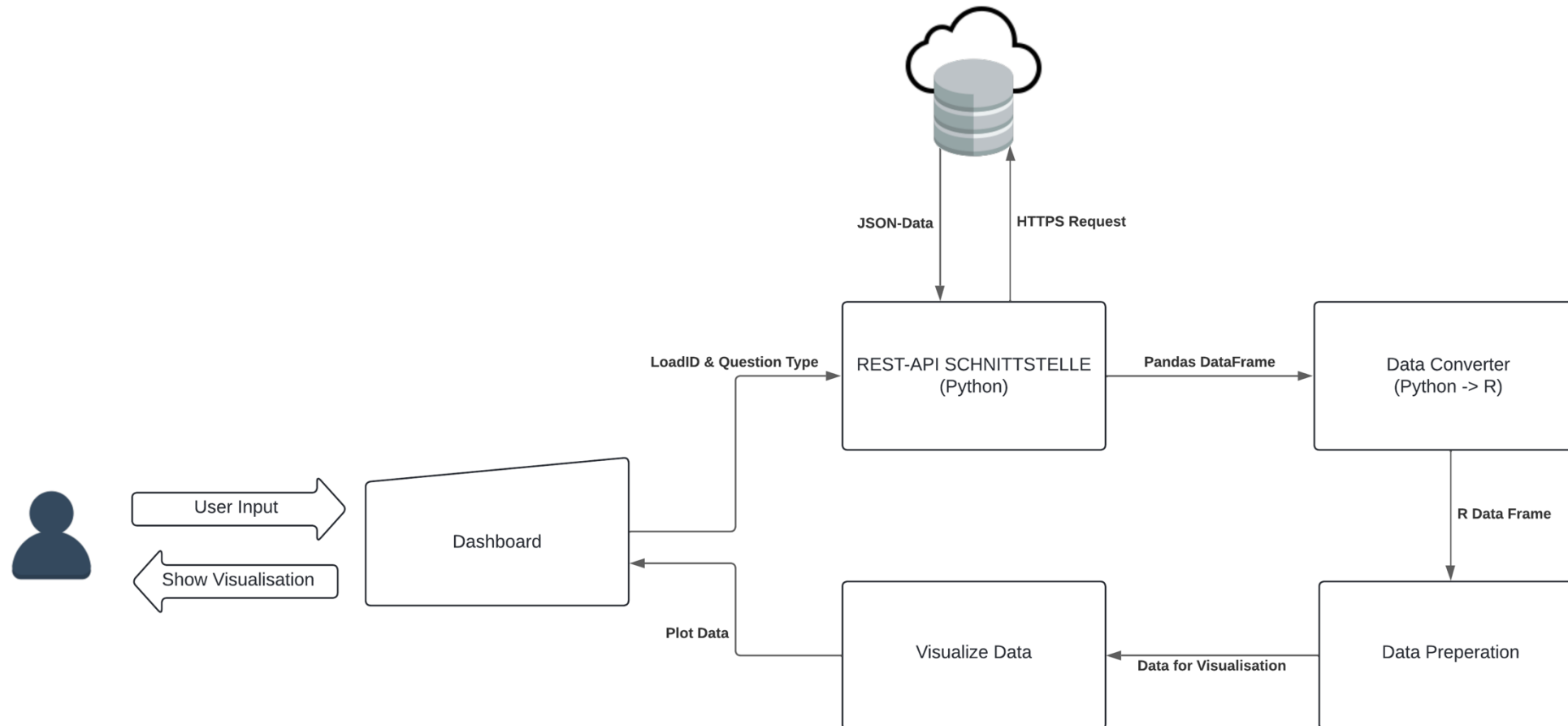


Views für einfacheren Zugriff auf Core

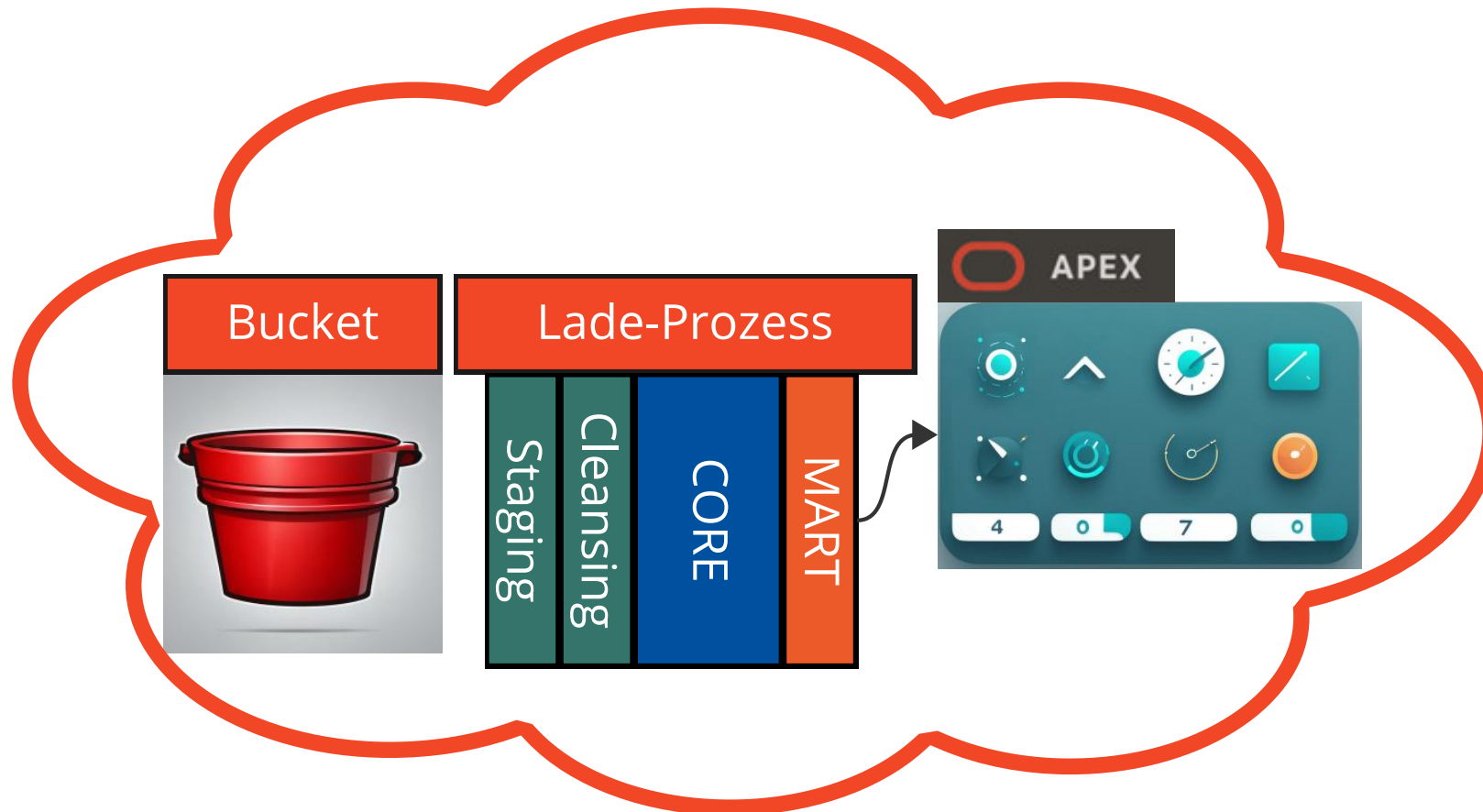
v_question_attempt



Dashboard versus APEX



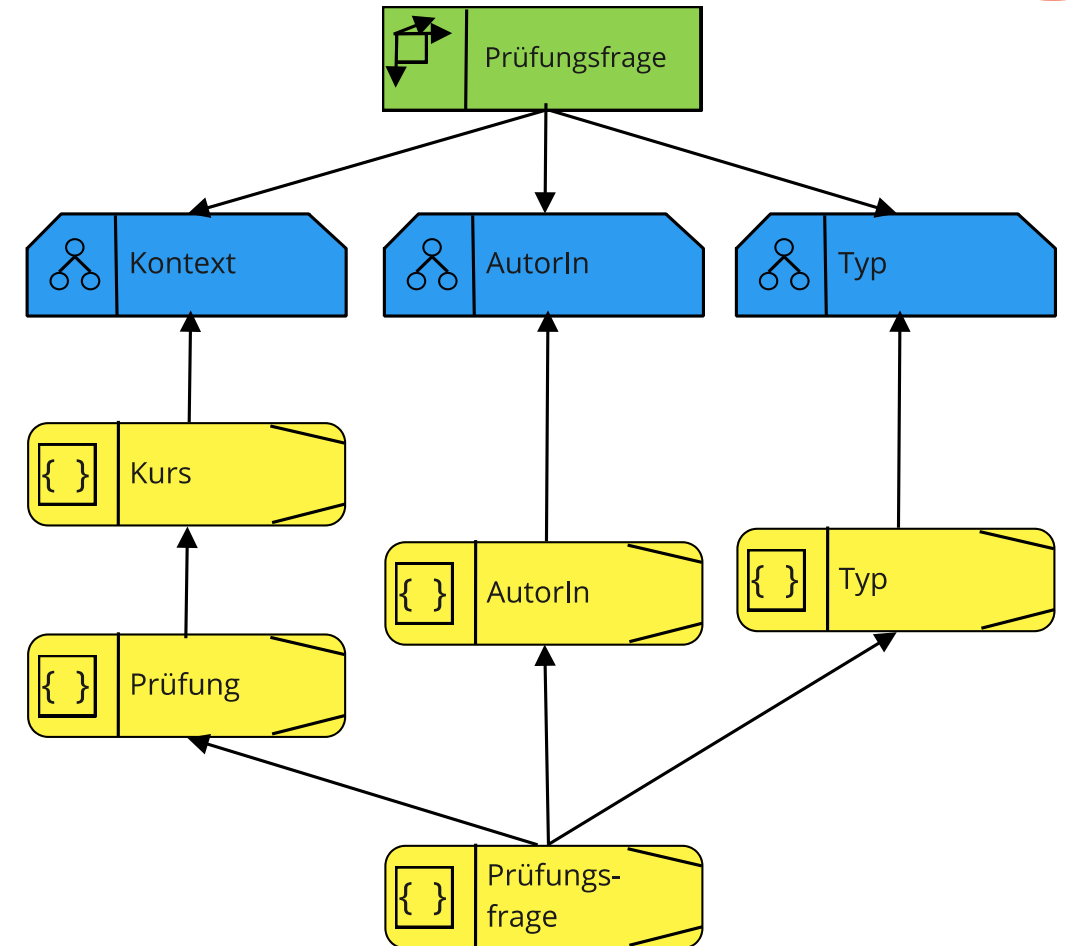
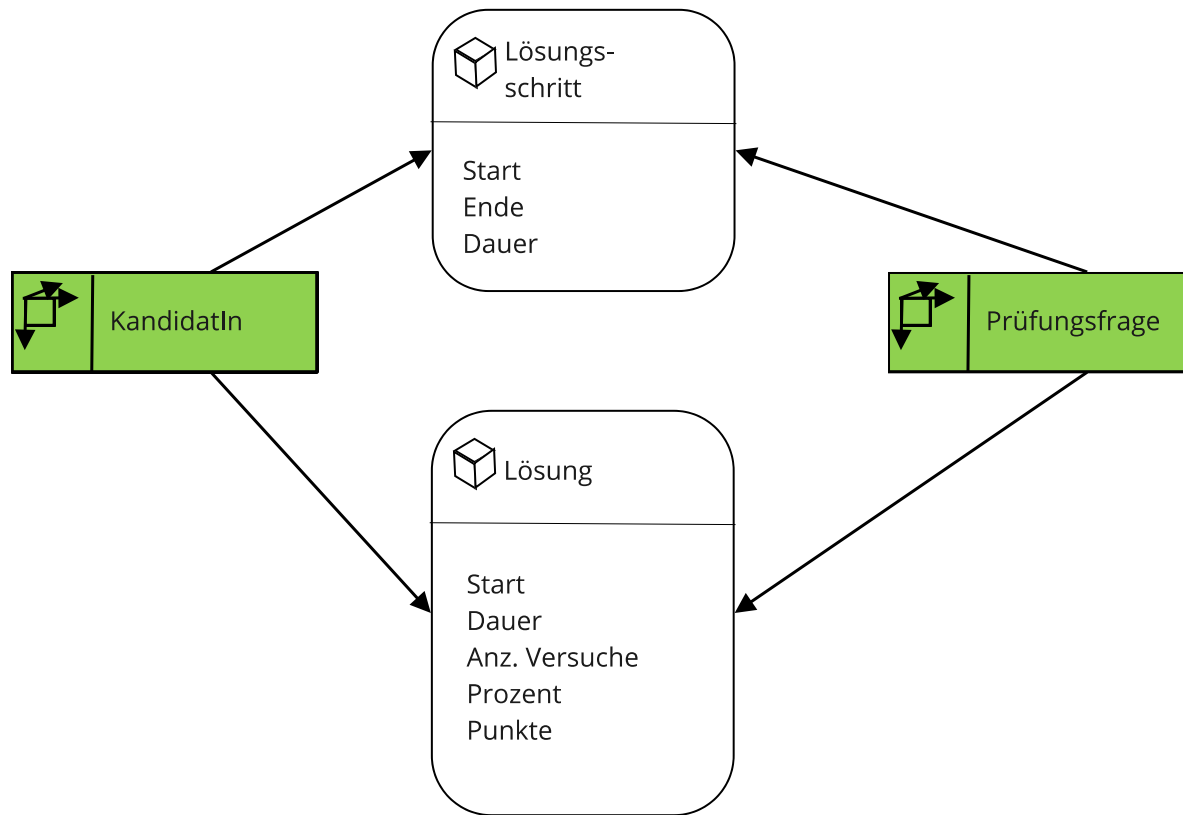
Dashboard versus APEX



Dashboard versus APEX

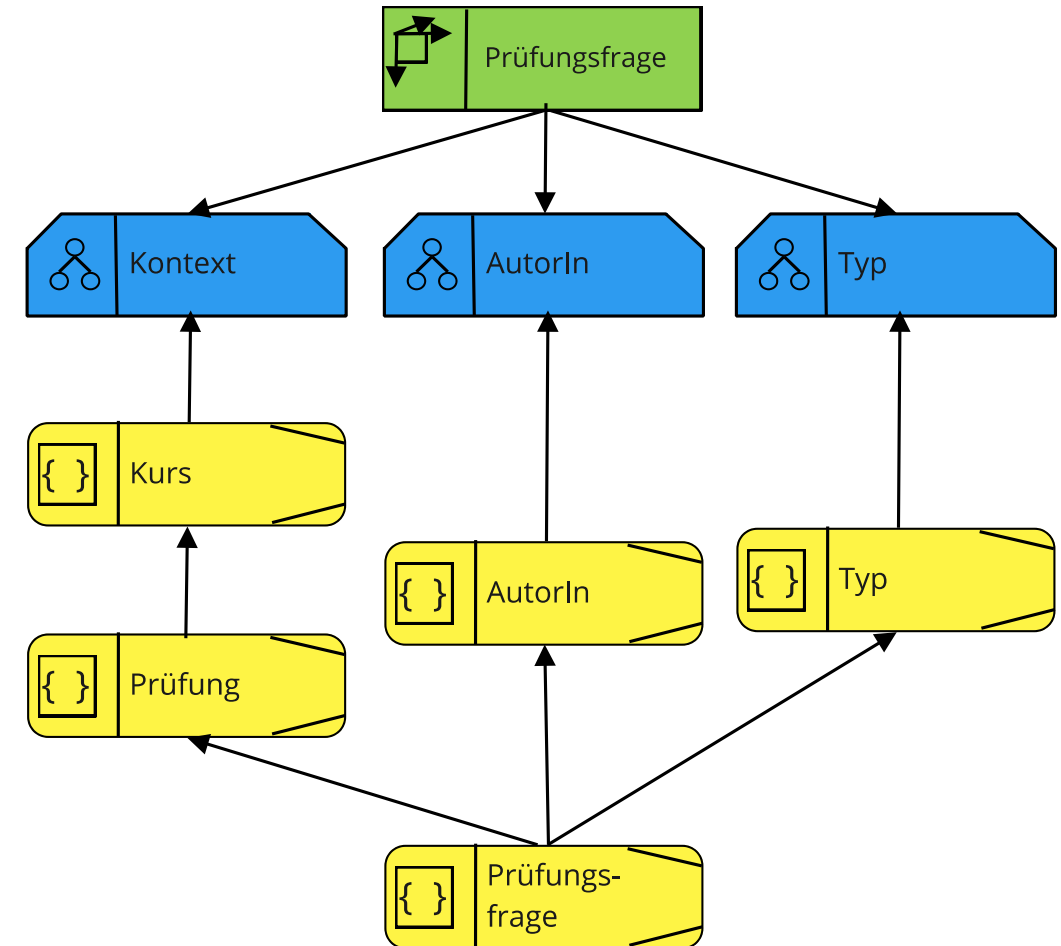
- Studis nutzen
R und Shiny Dashboard
- Daten direkt aus Core
- Datenanalyse mit R sehr flexibel
- Studis kennen R
- Via REST nach Dashboard
- APEX als Bestandteil der Cloud
- Daten in Mart
- Struktur muss vorher definiert werden
- Alles SQL
- Keine externe Schnittstelle

Erste Marts

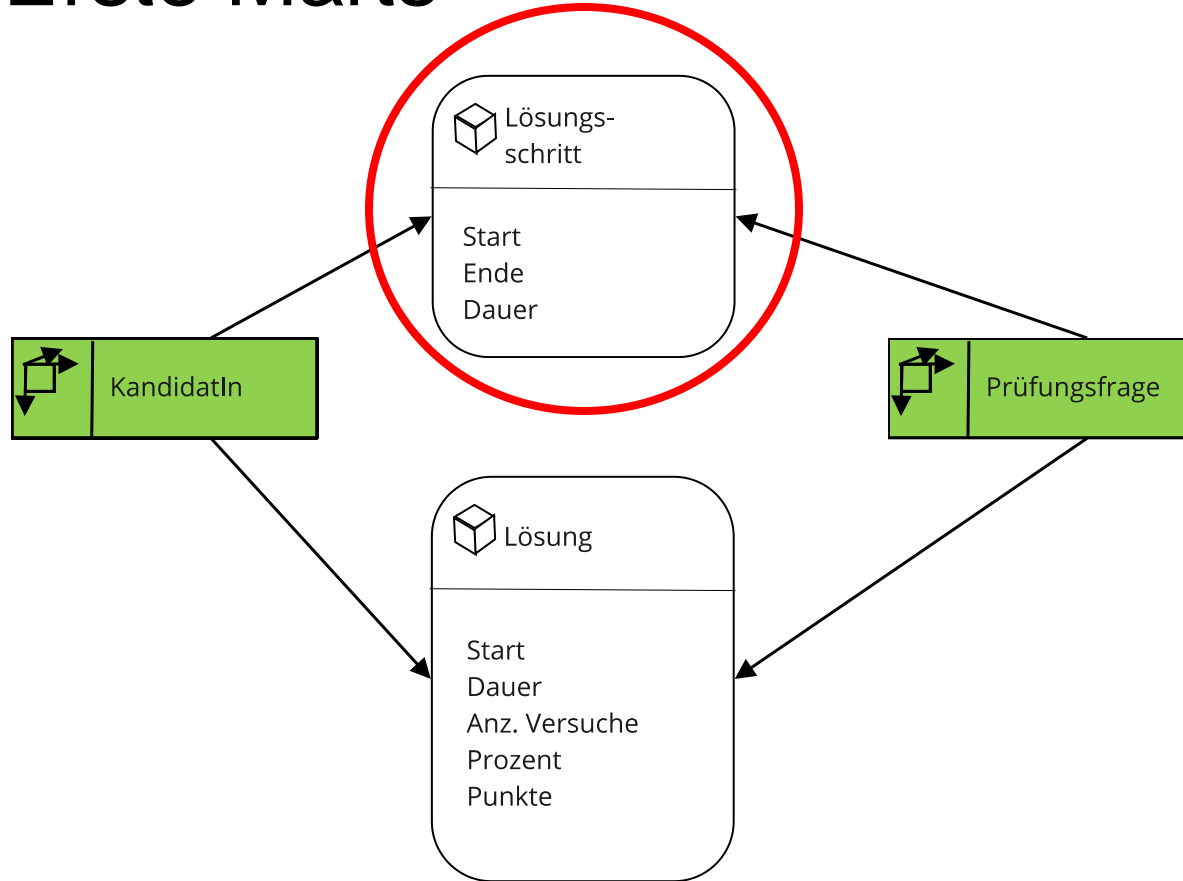


Vom Core in die Marts

```
INSERT INTO MD_MART.DM_D_Question
SELECT
    seq_md_id_question.nextval MD_ID_Question,
    q.question_id QU_MD_BK_question_id,
    q.load_id QZ_MD_BK_load_id,
    q.question_name QU_question_name,
    q.question_text QU_question_text,
    ...
    q.question_type TY_question_type
FROM MD_CORE.v_question q
WHERE q.load_id = this_load_id;
```



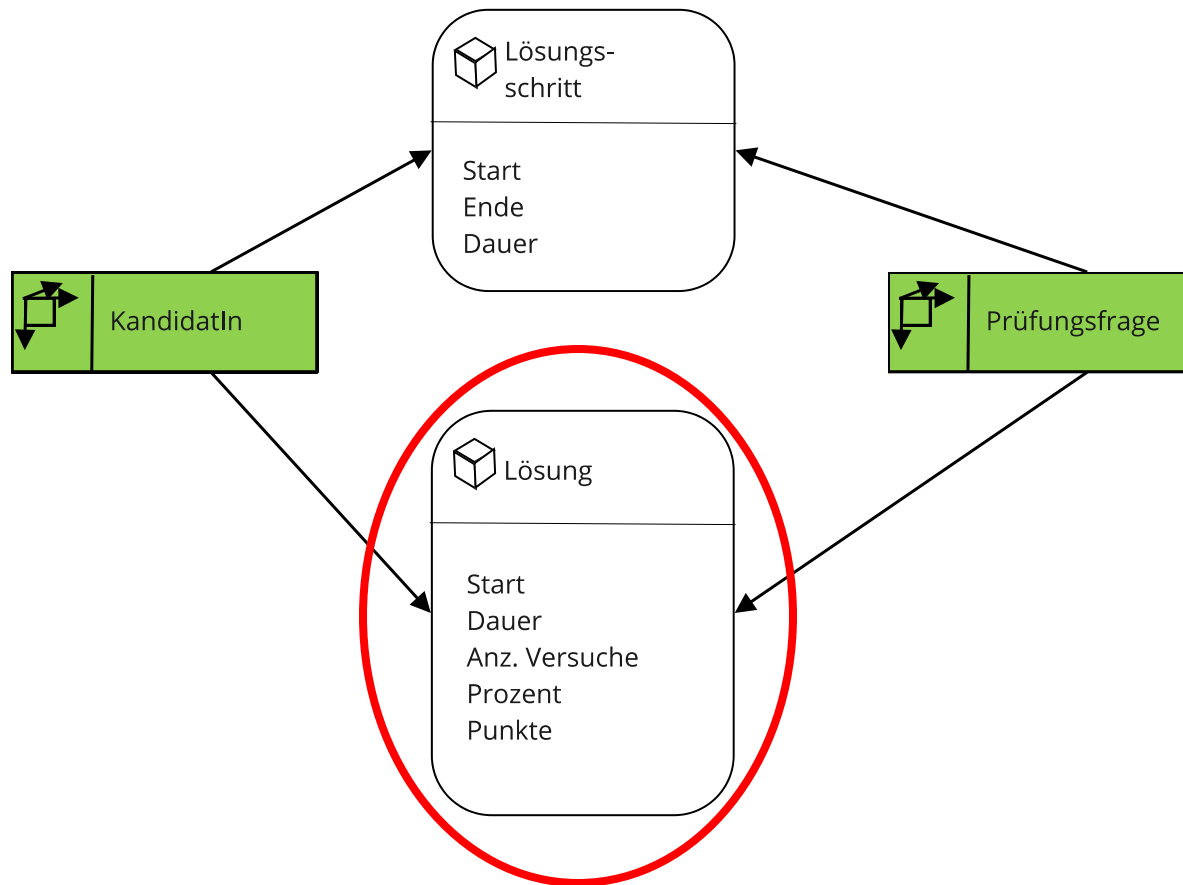
Erste Marts



Vom Core in die Marts

```
INSERT INTO MD_MART.DM_F_Question_Submission_Time_Segment
SELECT
    MD_ID_Question,
    quiz_attempt_user_hash MD_ID_Candidate,
    time_prev start_time,
    createdunixtime end_time, time_spent duration, load_id
FROM (
    SELECT
        d_q.MD_ID_Question,
        qa.quiz_attempt_user_hash, ...
        LAG(step.createdunixtime, 1, 0) OVER (PARTITION BY qa.QUIZ_ATTEMPT_USER_HASH
            ORDER BY step.createdunixtime) AS time_prev,
        step.createdunixtime -
            LAG(step.createdunixtime, 1, 0) OVER (PARTITION BY qa.QUIZ_ATTEMPT_USER_HASH
                ORDER BY step.createdunixtime) AS time_spent
    FROM md_core.V_QUESTION_ATTEMPT qa JOIN
        MD_CORE.V_QUESTION_ATTEMPT_STEP step
        ON qa.H_QUESTION_ATTEMPT_SID = step.H_QUESTION_ATTEMPT_SID JOIN
        ...
        MD_MART.DM_D_Question d_q ON d_q.QU_MD_BK_question_id = q.question_id AND d_q.QZ_MD_BK_load_id = q.load_id
    WHERE qa.load_id = this_load_id)
```

Erste Marts



Vom Core in die Marts

```
INSERT INTO MD_MART.DM_F_Question_Submission_Result
SELECT
    ts.MD_ID_Question MD_ID_Question,
    ts.MD_ID_Candidate MD_ID_Candidate,
    MIN(ts.start_time) attempt_start_time,
    SUM(duration) duration,
    COUNT(ts.md_id_question) number_of_visits,
    load_id load_id
FROM md_mart.dm_f_question_submission_time_segment ts
WHERE load_id = this_load_id
GROUP BY md_id_candidate, md_id_question, load_id;
```



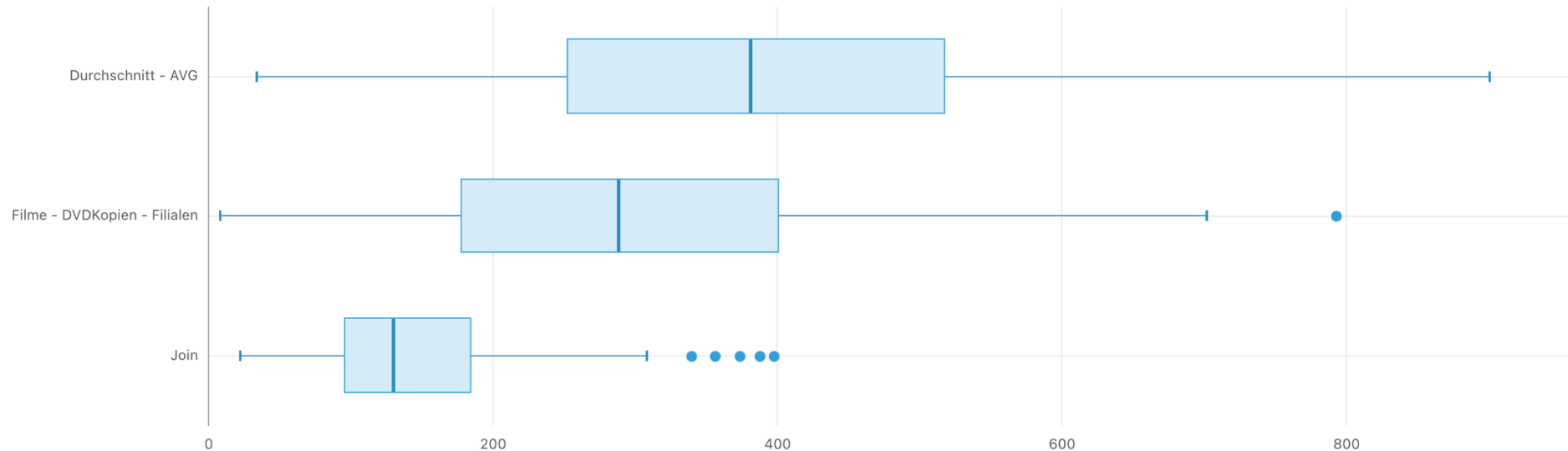
In APEX

≡ Attempt_Time

Overview





Zeitdauer pro Frage

22
22



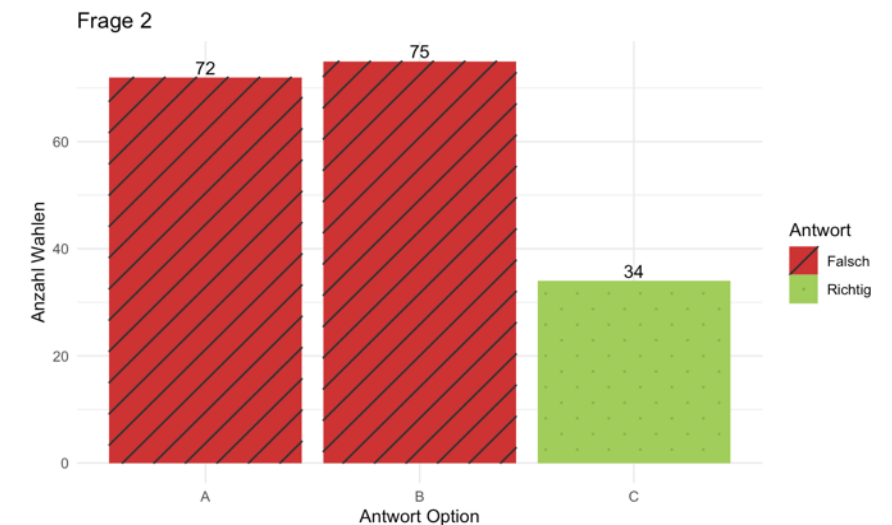
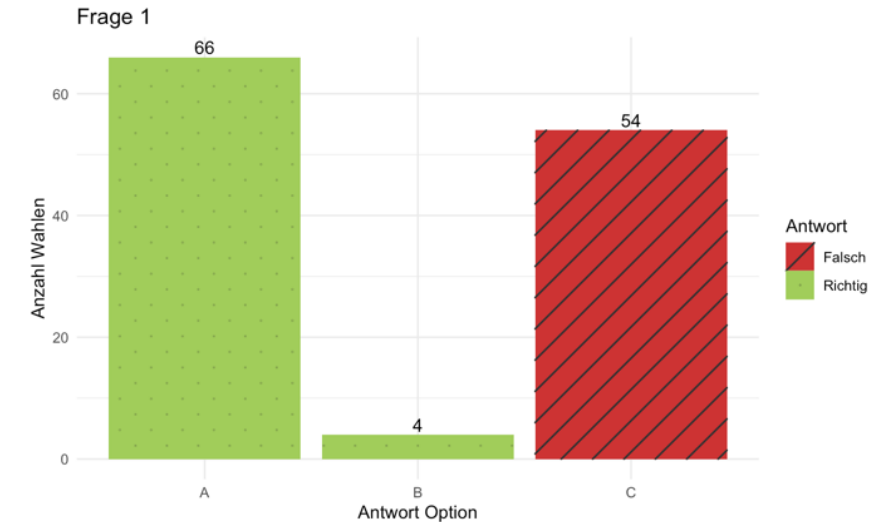
Erste Erweiterung: kprime und mtf

Markieren Sie den gewünschten Fragetyp ×

- ☐  Kprim (ETH)
- ☐  Lückentext (Cloze)
- ☐  Lückentextauswahl
- ☐  MTF (ETH)

Wählen Sie einen Fragetyp, um
seine Beschreibung zu sehen.

Problem:
Benötigen andere Daten aus anderen Quelltabellen



Stolpersteine

1. Mehrere Versionen von Daten
2. Staging neue Tabelle
3. Cleansing: Tabellen mit mehr Attributen
4. Core: Neue Satelliten
5. Mart: weitere Marts

Wie kann man das am einfachsten anpacken

Stolpersteine

1. Mehrere Versionen von Daten
 2. Staging neue Tabelle
 3. Cleansing: Tabellen mit mehr Attributen
 4. Core: Neue Satelliten
 5. Mart: weitere Marts
1. Attribute können NULL sein

Stolpersteine

1. Mehrere Versionen von Daten
2. Staging neue Tabelle
3. Cleansing: Tabellen mit mehr Attributen
4. Core: Neue Satelliten
5. Mart: weitere Marts

2. In Staging sind immer nur Daten vom aktuellen Load

Je Version eine Tabelle

View, die alle Tabellen mit UNION zusammenführt

Code Versionsabhängig

Stolpersteine

1. Mehrere Versionen von Daten
2. Staging neue Tabelle
3. Cleansing: Tabellen mit mehr Attributen
4. Core: Neue Satelliten
5. Mart: weitere Marts

3. In Cleansing sind immer nur Daten vom aktuellen Load

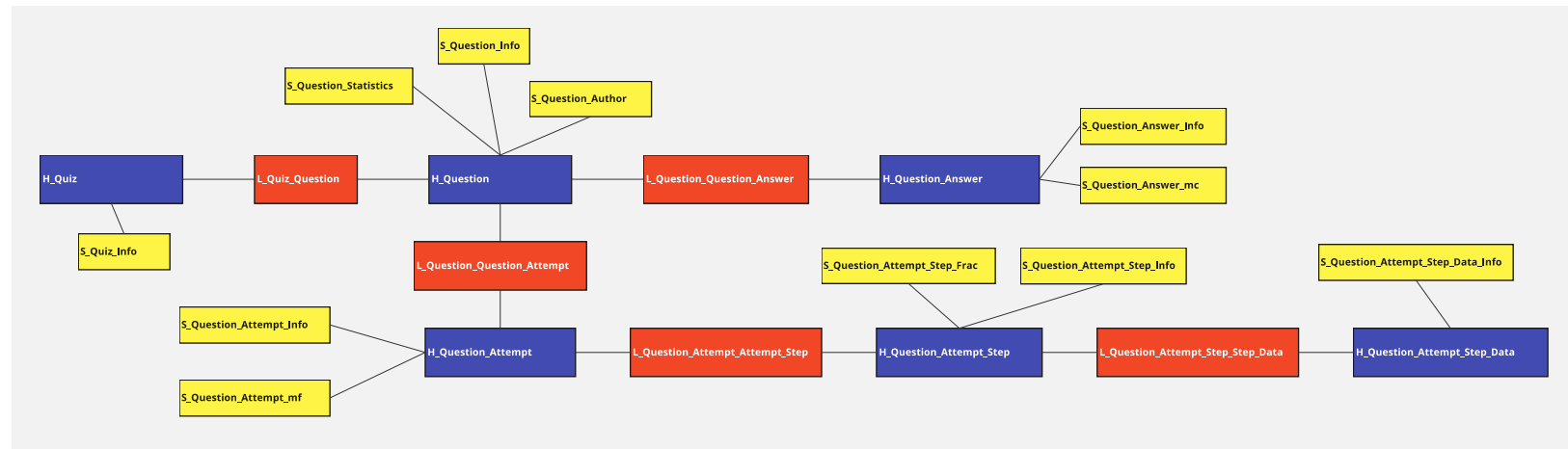
Tabellen bekommen mehr Attribute

Neue Tabellen

Code Versionsabhängig

Stolpersteine

1. Mehrere Versionen von Daten
 2. Staging neue Tabelle
 3. Cleansing: Tabellen mit mehr Attributen
 4. Core: Neue Satelliten
 5. Mart: weitere Marts
4. Core erweiterbar, da Data Vault



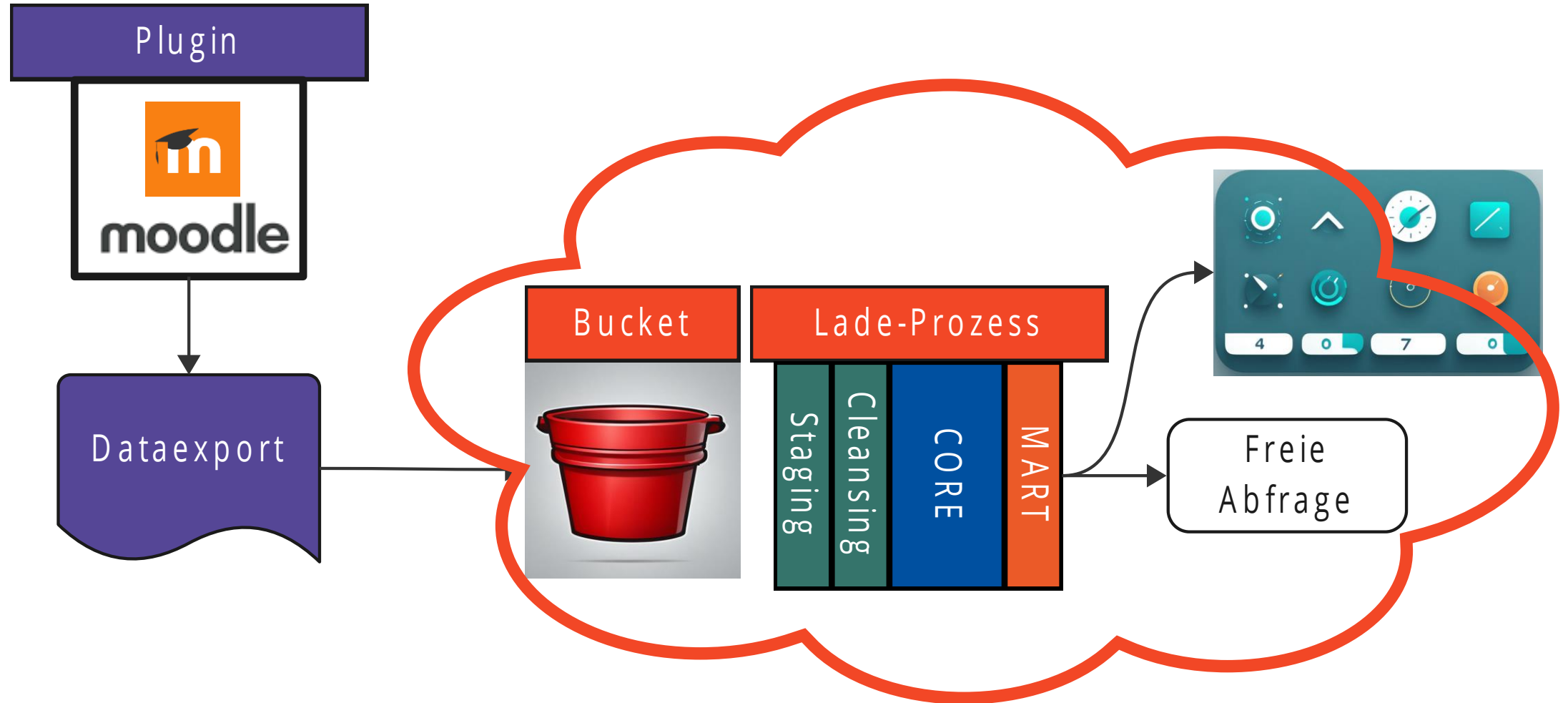
Flexible Lösungen

1. View für Staging
2. Daten in Cleansing immer löschen
3. Core als Data Vault
4. Views auf Core
5. IF (NOT) EXISTS in Skripten

Was kommt als nächstes?

1. Auswertung Multiple Choice
2. Weitere Fragetypen
3. Quiz umsetzen
4. Mehr als eine Prüfung exportieren

Übersicht



Fazit

1. Pilot bringt Mehrwert
2. Einfach ausbaubar
3. Klare Konzepte sind flexibel

Dr. Andrea Kennel

Blogbeitrag auf Hochschulforum Digitalisierung:

<https://hochschulforumdigitalisierung.de/learning-analytics-ermöglichen>



andrea.kennel@fhnw.ch

andrea@infokennel.ch

www.infokennel.ch