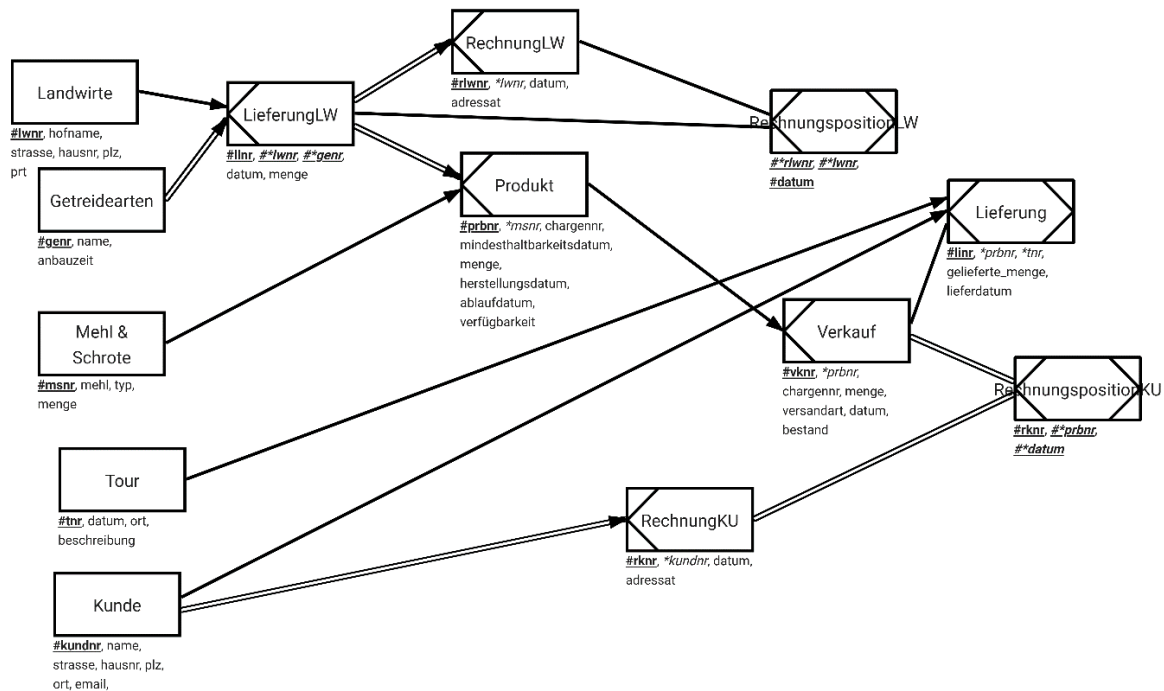


Mühle Datenmodell – Gruppe 1.5



Dies ist das alte Datenbankmodell, welches für die erste Abgabe erstellt wurde. Hierbei wird das Modell in zwei geteilt und durch die Tabellen Produkt und Verkauf zusammengesetzt. Der erste Teil befasst sich mit der Produktion und der zweite mit der Lieferung.

Landwirte liefern verschiedene **Getreidearten** an das Unternehmen, die durch die Entität **LieferungLW** erfasst werden. Jede Lieferung beinhaltet Informationen wie Datum und Menge. Das Getreide wird dann zu **Mehl & Schrote** verarbeitet, wobei diese Entität Details wie Typ und Menge enthält.

Die daraus entstehenden Produkte werden in der Entität **Produkt** erfasst, welche Attribute wie Chargennummer, Mindesthaltbarkeitsdatum und Verfügbarkeit enthält. Produkte werden an Kunden verkauft, was durch die Entität **Verkauf** abgedeckt ist. Diese beinhaltet Informationen über Menge, Versandart und Bestandsdaten.

Lieferungen an Kunden werden durch die Entität **Lieferung** dokumentiert, einschließlich des Lieferdatums und der gelieferten Menge. Kundenrechnungen und deren Positionen werden durch die Entitäten **RechnungKU** und **RechnungspositionKU** verwaltet, während Landwirte Rechnungen über **RechnungLW** und **RechnungspositionLW** erhalten.

Zusätzlich gibt es die Entität **Tour**, die Lieferungen und Besuche bei Kunden organisiert, einschließlich Details wie Datum und Ort.

Dieses Modell ermöglicht eine detaillierte und strukturierte Verwaltung der gesamten Lieferkette vom Landwirt bis zum Endkunden.

Query 1:

```
/*Wie viele Tonnen Weizenmehl Type 405 wurden im März 2022 verkauft?*/
SELECT SUM(v.menge) / 1000 AS tonnen
FROM verkauf v
JOIN produkt p ON v.prbnr = p.prbnr
JOIN mehl_und_schrote ms ON p.msnr = ms.msnr
WHERE ms.mehl = 'Weizenmehl' AND ms.typ = 'Typ 405'
AND v.datum BETWEEN '2022-03-01' AND '2022-03-31';
```

Ergebnis 1:

Ergebnis Nr. 1 (+)

tonnen
60

Query 2:

```
/*Welcher Landwirt hat 2021 am meisten Dinkel geliefert?*/
SELECT l.hofname, SUM(lw.menge) AS menge
FROM lieferung_lw lw
JOIN landwirt l ON lw.lwnr = l.lwnr
JOIN getreidearten g ON lw.genr = g.genr
WHERE g.name = 'Dinkel' AND lw.datum BETWEEN '2021-01-01'
AND '2021-12-31'
GROUP BY l.hofname
ORDER BY menge DESC
LIMIT 1;
```

Ergebnis 2:

Ergebnis Nr. 1 (+)

hofname	menge
Hof Keller	2300

-1 of< < > >|

Query 3:

```
/*Wie viel Roggenmehl Type 1150 ist derzeit auf Lager?*/
SELECT SUM(p.menge) AS lagerbestand
FROM produkt p
JOIN mehl_und_schrote ms ON p.msnr = ms.msnr
WHERE ms.mehl = 'Roggenmehl' AND ms.typ = 'Typ 1150'
AND p.verfugbarkeit = TRUE;
```

Ergebnis 3:

Ergebnis Nr. 1 (+)

lagerbestand
2050

Query 4:

```
/*In Charge 20220425WZ550 wurden Mehlwürmer entdeckt. Wi
viel Mehl wurde hiervon bereits verkauft?*/
SELECT SUM(v.menge) AS verkauft
FROM verkauf v
JOIN produkt p ON v.prbnr = p.prbnr
WHERE p.chargenr = '20220425WZ550';
```

Ergebnis 4:

Ergebnis Nr. 1 (+)

verkauft
90

Query 5:

```
/*Welche Produkte sind im Lager vorhanden und verfügbar?*/
SELECT p.prbnr, ms.mehl, ms.typ, p.menge
FROM produkt p
JOIN mehl_und_schrote ms ON p.msnr = ms.msnr
WHERE p.verfugbarkeit = TRUE;
```

Ergebnis 5:

Ergebnis Nr. 1 (+)

prbnr	mehl	typ	menge
3	Gerstenschrot	Schrot	75
4	Hafermehl	Typ 630	125
5	Maismehl	Fein	60
6	Dinkelmehl	Typ 630	110
7	Hirsemehl	Fein	90
8	Reismehl	Fein	130
9	Sorghummehl	Fein	80
10	Amaranthmehl	Fein	140
11	Weizenmehl	Typ 405	500

Query 6:

```
/*Welche Landwirte haben mehr als 1000 kg Weizen
geliefert?*/
SELECT l.hofname, SUM(lw.menge) AS menge
FROM lieferung_lw lw
JOIN landwirt l ON lw.lwnr = l.lwnr
JOIN getreidearten g ON lw.genr = g.genr
WHERE g.name = 'Weizen'
GROUP BY l.hofname
HAVING SUM(lw.menge) > 1000;
```

Ergebnis 6:

Ergebnis Nr. 1 (+)

hofname	menge
Hof Bauer	1700
Hof Keller	1900
Hof Meier	1600
Hof Müller	1500
Hof Schmidt	1300

-5 of< < > >|

Query 7:

```
/*Welche Kunden haben Produkte aus der Charge CH010
gekauft?*/
SELECT DISTINCT k.name, k.adresse, k.email
FROM verkauf v
JOIN produkt p ON v.prbnr = p.prbnr
JOIN kunde k ON v.vknr = k.kundnr
WHERE p.chargenr = 'CH010';
```

Ergebnis 7:

Ergebnis Nr. 1 (+)

name	adresse	email
Laura Beispiel	Beispielallee 10	laura@beispiel.de

1 of 1 |< < > >|

TRIGGER:

```
/*Trigger zur Überprüfung des Mindestbestands beim
Verkauf*/
CREATE OR REPLACE FUNCTION pruefe_mindestbestand() RETURNS
TRIGGER AS $$
DECLARE
    lagerbestand INTEGER;
BEGIN
    SELECT menge INTO lagerbestand FROM produkt WHERE
prbnr = NEW.prbnr;
    IF lagerbestand - NEW.menge < 50 THEN
        RAISE EXCEPTION 'Mindestbestand unterschritten';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER mindestbestand_trigger
BEFORE INSERT OR UPDATE ON verkauf
FOR EACH ROW
EXECUTE FUNCTION pruefe_mindestbestand();
```

Query 8:

```
/*Wie viele verschiedene Kunden haben im Jahr 2022
eingekauft?*/
SELECT COUNT(DISTINCT v.vknr) AS kundenanzahl
FROM verkauf v
WHERE v.verkauft BETWEEN '2022-01-01' AND '2022-12-31';
```

Ergebnis 8:

Ergebnis Nr. 1 (+)

kundenanzahl
5

```
INSERT INTO verkauf (prbnr, verkauft, menge, versandart,
datum, bestand) VALUES
(1, '2023-02-15', 60, 'Standard', '2023-02-10', 40);
```

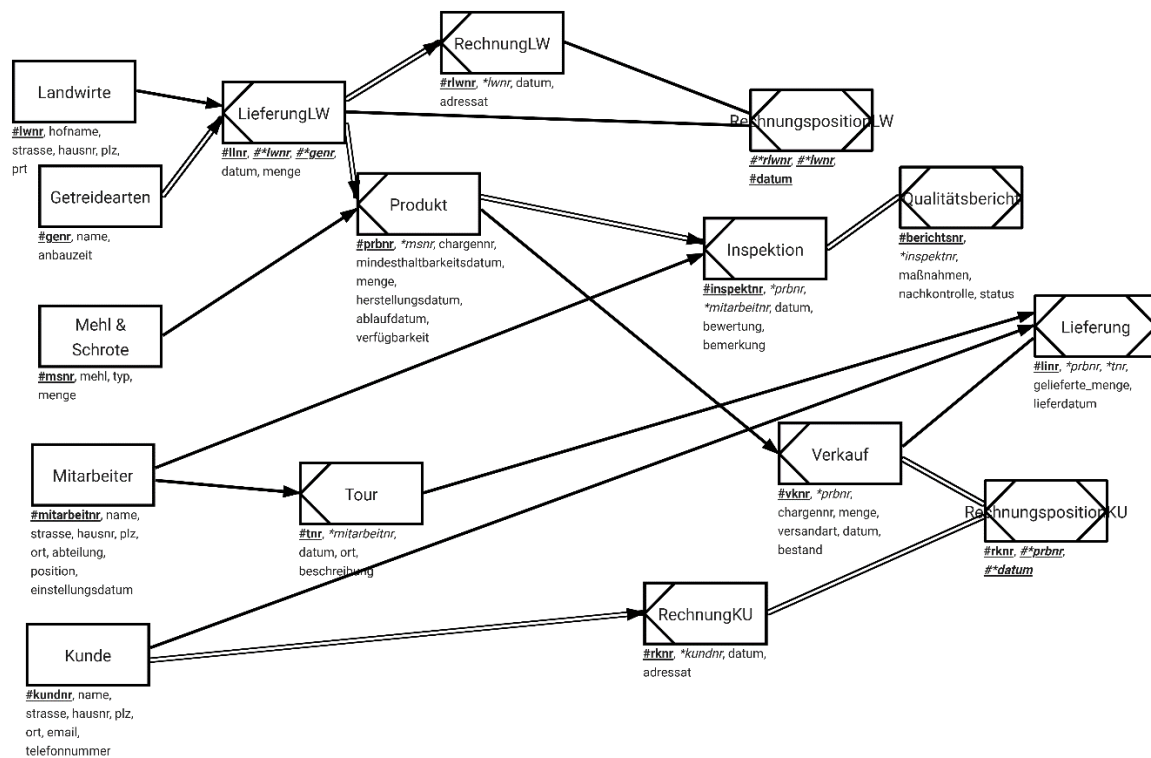
Ergebnis Nr. 1 (+)

✗ SQL Error ✗
Please check your input

Es gab eine SQLException: ERROR: Mindestbestand unterschritten Where: PL/pgSQL function pruefe_mindestbestand() line 9 at RAISE

PROBLEME UND ERWEITERUNGSMÖGLICHKEITEN:

- Produktionsprozesse sind nicht detailliert genug dargestellt (Schritte zwischen der Anlieferung des Getreides und der Produktion des Endprodukts.)
 - o Einführung von Entitäten wie Produktionsschritt (mit Attributen wie ProduktionsschrittID, Beschreibung, Dauer) und Verknüpfung mit Produkt und Mehl & Schrote
- Modell berücksichtigt nicht alle möglichen Interaktionen mit Kunden (Rücksendungen und Kundenfeedback)
 - o Einführung einer Entität Rücksendung (mit Attributen wie RücksendungsID, Datum, Grund, Menge) und Kundenfeedback (mit Attributen wie FeedbackID, KundeID, Kommentar, Bewertung)
- Fehlen der Mitarbeiter, die in Produktion, Lieferung arbeiten
 - o Einführung einer Entität Mitarbeiter mit Attributen wie MitarbeiterID, Name, Position und Kontaktinformationen. Diese Entität könnte mit Produktion und Lieferung verknüpft werden, um zu verfolgen, welcher Mitarbeiter für welche Aufgaben zuständig ist
- Berücksichtigt keine Inspektionen, die für die Qualitätssicherung und Einhaltung von Standards in den Produktions- und Lieferprozessen entscheidend sind
 - o Einführung einer Entität Inspektion mit Attributen wie InspektionsID, Datum, Ergebnis, Bemerkungen



Die Entität **Mitarbeiter** wurde hinzugefügt, um alle Mitarbeiter in der Mühle zu erfassen und ihnen ihre Bereiche (Produktion, Qualitätssicherung, Lieferung) zuzuordnen. Dies ermöglicht eine präzisere Verfolgung von Arbeitskräften und deren Verantwortlichkeiten.

Die **Tour**-Entität wurde erweitert, um den Touren ein Lieferant zuzuordnen.

Zusätzlich wurde die Entität **Inspektion** hinzugefügt, um Inspektionen der Produkte zu dokumentieren. Diese Entität umfasst Details wie das Datum der Inspektion, die Bewertung, Bemerkungen sowie die Zuordnung der verantwortlichen Mitarbeiter und Produkte. Dies stellt sicher, dass alle Inspektionen systematisch erfasst und ausgewertet werden können.

Zur weiteren Qualitätssicherung wurde die Entität **Qualitätsbericht** eingeführt. Diese enthält Informationen über die Maßnahmen, die aus den Inspektionen resultieren, die Nachkontrolle und den aktuellen Status der Qualitätsprüfungen. Dies ermöglicht eine umfassende Übersicht über die Qualitätssicherung und die durchgeführten Maßnahmen.

Insgesamt ermöglicht das erweiterte Modell die Verwaltung der Mitarbeiter, die Dokumentation von Inspektionen und die Qualitätssicherung, was zu einer verbesserten Effizienz und Transparenz im Unternehmen führt.