Here, I wanna completely design a data warehouse and make the relational database of no use to the analysts. This way, I increase the redundancy, but I also increase the read speed by denormalizing:

All the first IDs in each table are the unique identifiers that the system creates. I treat the IDs of the OLTP system as a business key. All the relations through these facts and dims exist through the unique IDs of the data warehouse and not the OLTP IDs (surrogate key).

- **dim_region:**
    - region_id:
    - region_name

- **dim_customer**
    - customer_id :
    - customer_bk: kdnr
    - verlagsname
    - region_id

- **dim_status**
    - status_id
    - status_name (values explained in the ETL fix)

- **dim_content**
    - content_id
    - content_bk

- **dim_date**
    - date_id
    - date_full
    - year
    - month
    - day

        - (Other date formats needed based on the business)

- **fact_invoice_position** (to remove the join and increase the read)
    - invoice_position_id
    - invoice_bk
    - position_bk
    - customer_id
    - status_id
    - content_id
    - redatum_date_id
    - zahlungsdatum_date_id
    - verdatum_date_id
    - nettobetrag

- **fact_invoice**
    - invoice_id
    - invoice_bk
    - customer_id
    - status_id
    - redatum_date_id
    - zahlungsdatum_date_id
    - position_count
    - position_nettobetrag_sum
    - summenetto
    - mwstsatz
    - summenebenkosten
    - zahlungsbetragbrutto
    - is_date_conflict (if redatum < zahlungsdatum)
    - is_orphan
    - is_customer_conflict (if more than one customer is assigned to this invoice)
    - is_amount_conflict (if zahlungsdatum has value but no zahlungsbetragbrutto or the calculation between tax or additional cost won't match zahlungsbetragbrutto)
    - is_invoice_position_amount_conflict (if the amount of the invoice wont match the sum of the amounts of the linked positions)

- **fact_position**

- position_id
- position_bk
- invoice_bk
- customer_id
- content_id
- verdatum_date_id
- invoice_status_id
- nettobetrag
- is_date_conflict (if verdatum is null or greater than zahlungsdatum)
- is_orphan (if the position is linked to no invoice)

- **fact_aggregated_measure_nonconformity**

    - id
    - date_id
    - invoice_with_more_than_one_customer_count
    - invoice_position_conflict_amount
    - invoice_date_conflict_count
    - position_date_conflict_count
    - invoice_position_date_conflict_count

I can also propose other facts and dims to the users like customer rfm, cohort, real payment, customer behavior analysis, etc.