

LUNAR PROSPECTORS

SCIENCE OF TOMORROW, TODAY!



Verwaltung einer bemannten Mondbasis für Forschung und Wirtschaft

Ben Winter
Tom Schröder
Marcel Otzen

Sprecher: Ben Winter
E-Mail: ben_niklas.winter@hsbi.de

Abschlusspräsentation

GLIEDERUNG



- Firmenkonzept
- Use-Cases
- Geschäftsbereiche
- ERM
- PDM
- Beispieldatengenerierung
- Live SQL Demo
- Aussichten

FIRMENKONZEPT



Vollständige Verwaltung und Steuerung einer dauerhaft bewohnten Station für Forschungsmissionen oder wirtschaftliche Operationen auf der Mondoberfläche

- Verwaltung aller Personen der Mondbasis (mit Qualifikation, Ressourcenbedarf, Termine)
- Verwaltung aller Ressourcen, Gerätschaften und Ersatzteile
- Organisation von Versorgungslieferungen (für Ressourcen, Ersatzteile sowie Personentransport)
- Organisation (Team und Ressourcenbedarf) von Mondmissionen für Forschung, Ressourcengewinn oder Anlagen-Wartung bzw. -Ausbau
- Verwalten aller Termine (Lieferungen, Missionen, etc.) der Station



USE-CASES



- Station für staatliche und privatwirtschaftliche Forschung (z.B. für NASA-Artemis)
- „Weltraumbahnhof“ zur Neubefüllung von Raketen für längere Reisen z.B. zum Mars

GESCHÄFTSBEREICHE



- Ressourcenverwaltung (z. B. Lebensmittel, Ersatzteile, Geräte)
 - Alle Lagerbestände verwalten
 - Ressourcen verwalten
 - Planen von Versorgungslieferungen, Aufgeben der Bestellungen und Organisation von Personentransporten
- Arbeitseinteilungen für alltägliche Aufgaben
 - Verwaltung und Planung aller Jobs (laufend / noch ausgeschrieben), inklusive genauer Arbeitszeiten, Treffpunkte und benötigter Geräte und Ressourcen
 - Verwaltung der Personen samt deren Qualifikationen
 - Bewerbungen und entsprechende Jobzuweisungen verwalten
- Missionsverwaltung
 - Verwalten aller Missionen (laufend / geplant) mit beteiligten Gruppen und deren Jobs
 - Verwaltung der Bestände (Ressourcen und Geräte) mit auf Mission

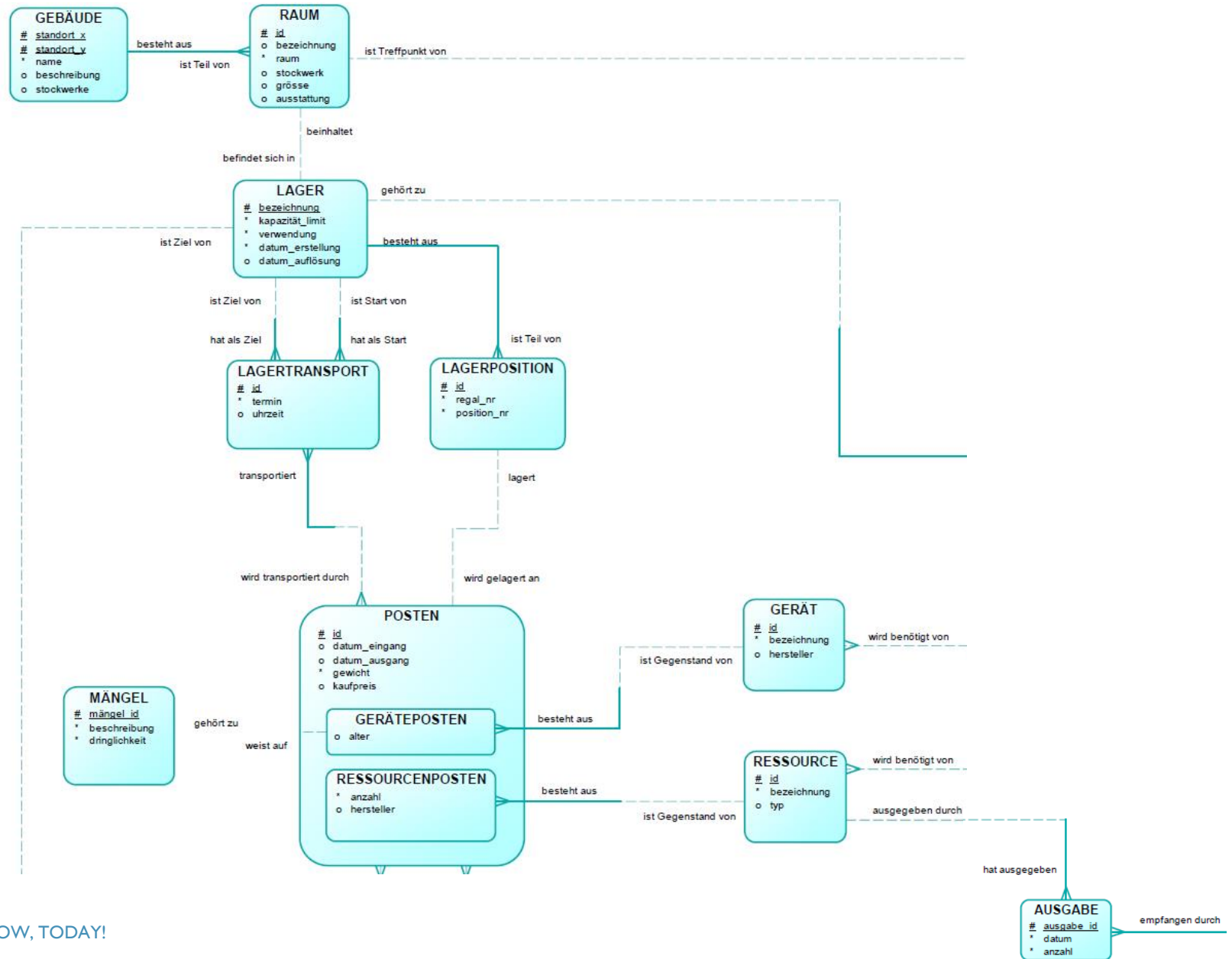


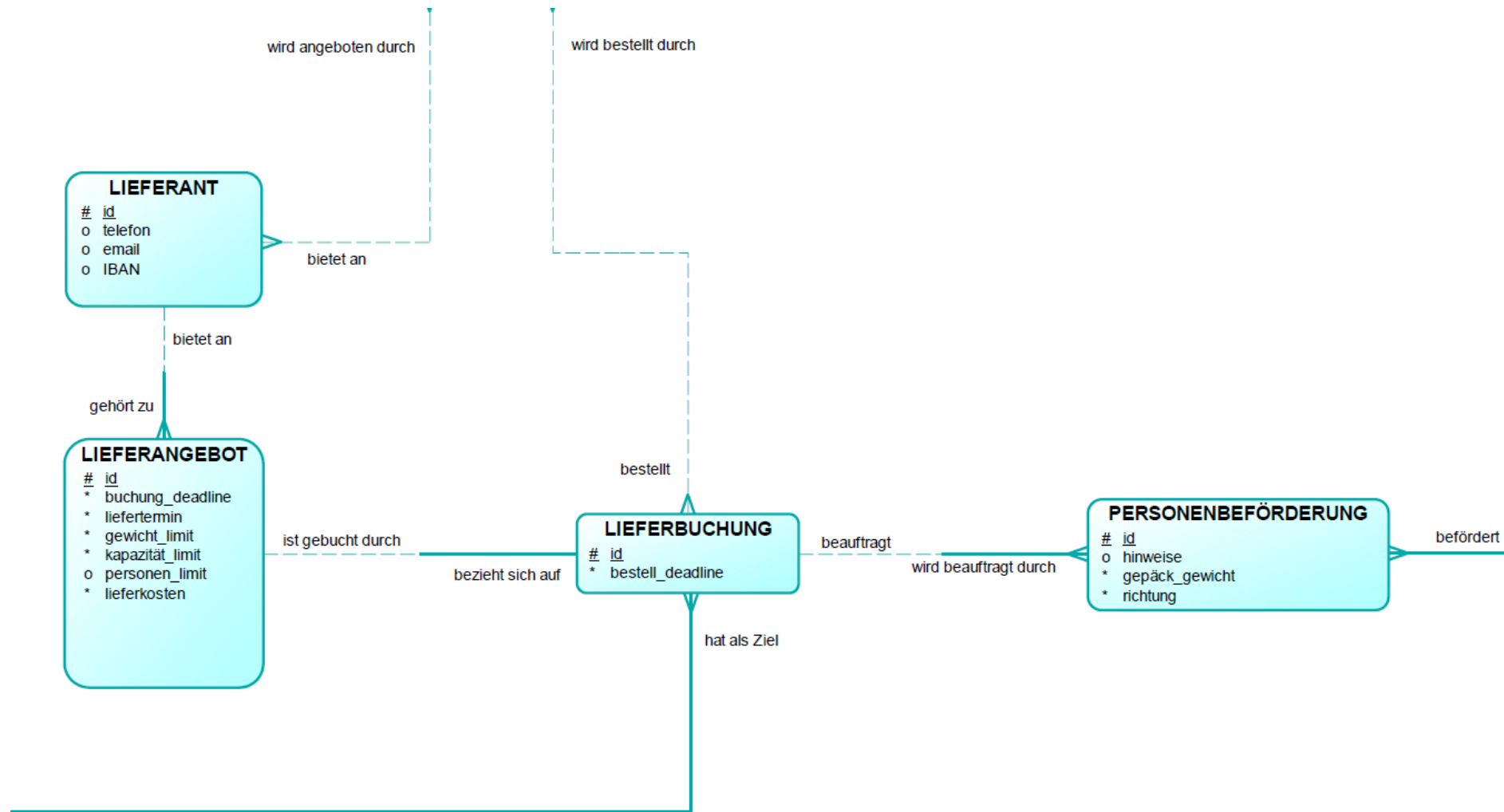
Geschäftsbereich

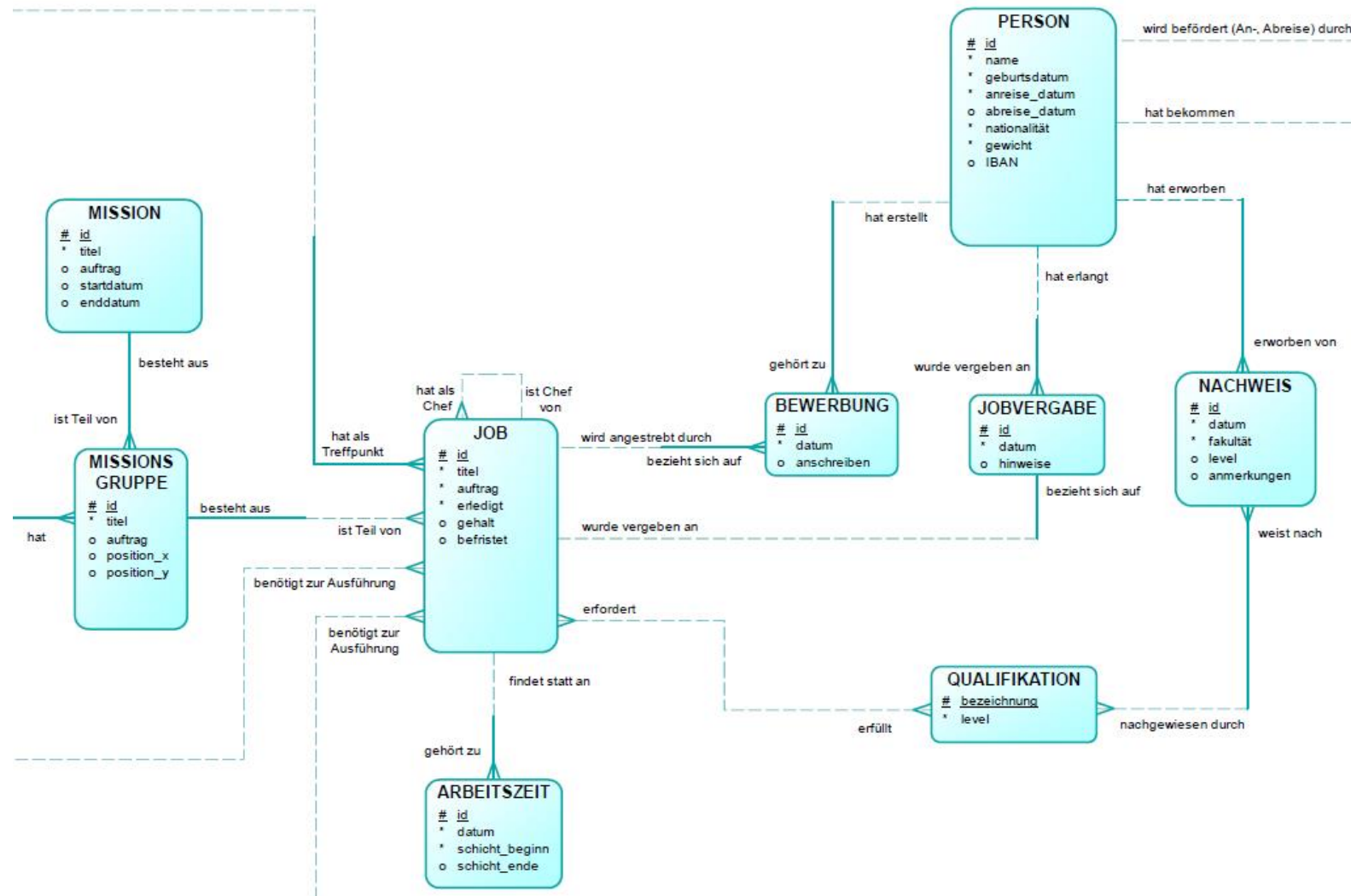
Missionsverwaltung

Teilbereich

Arbeitseinteilung







PHYSISCHES DATENMODELL



Geschäftsbereich

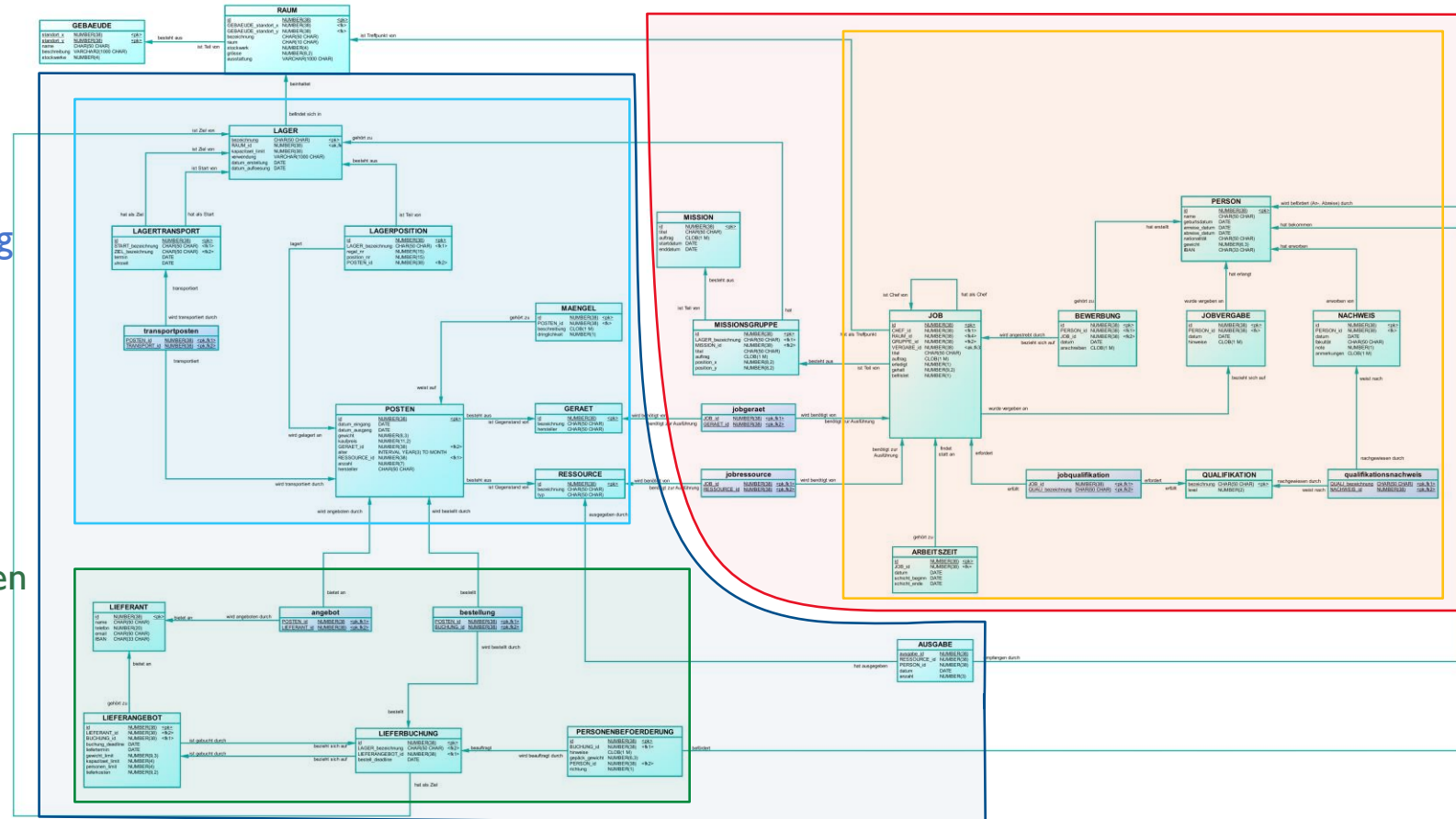
Ressourcenverwaltung

Teilbereich

Lagerbestände

Teilbereich

Versorgungslieferungen



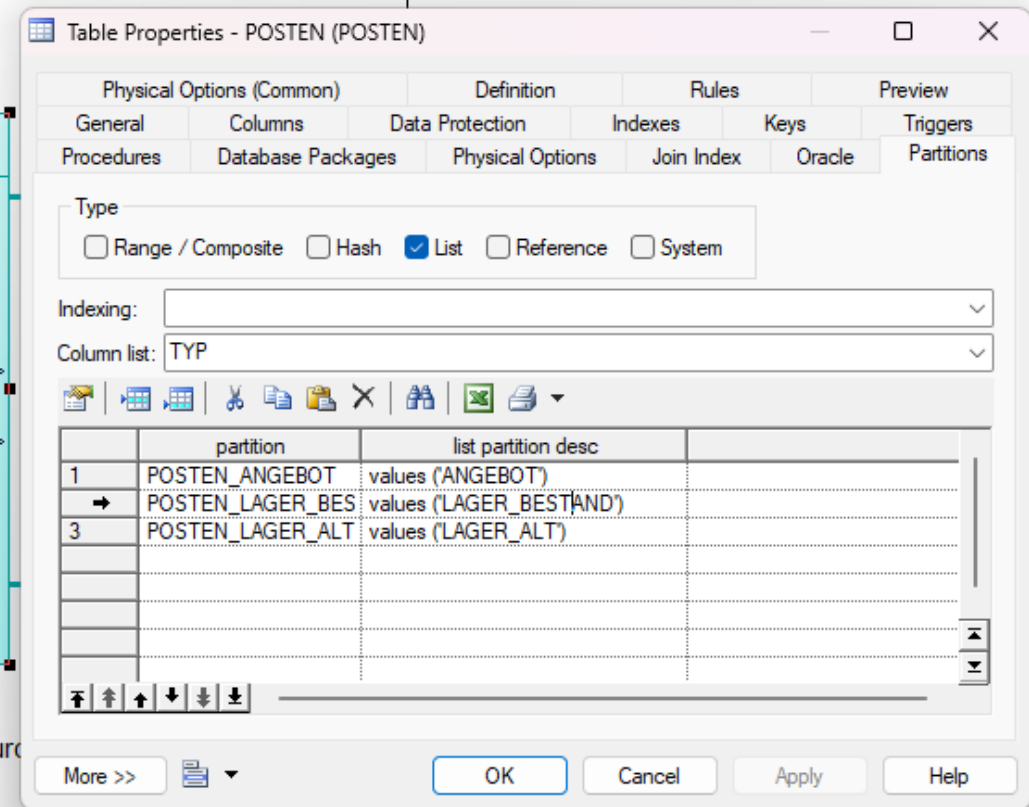
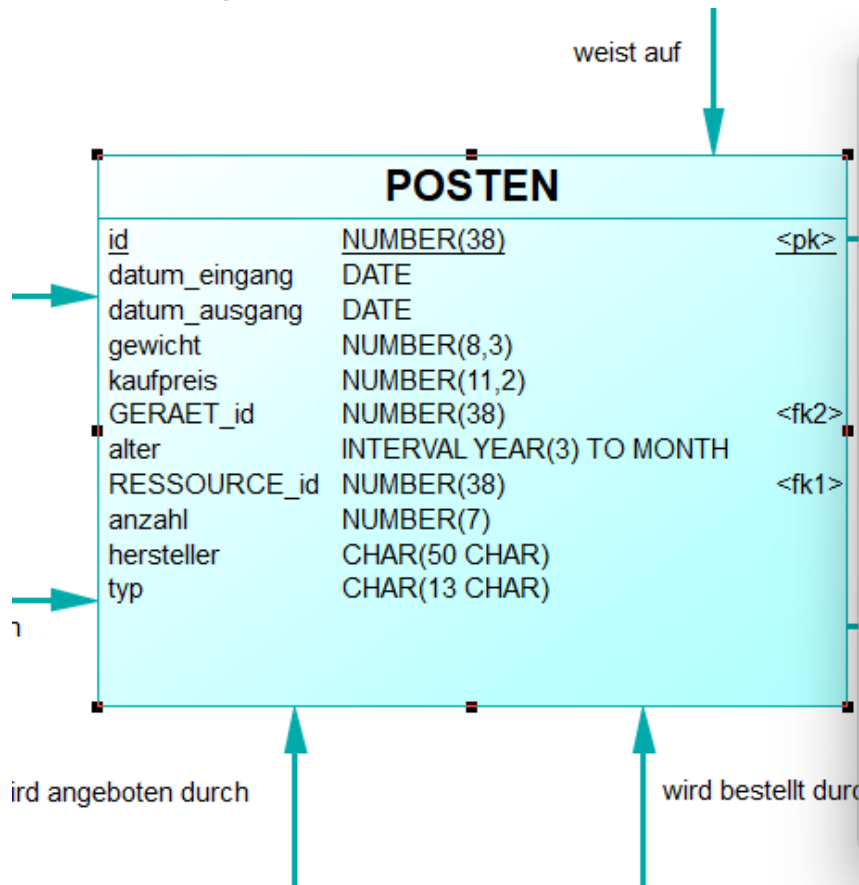
Geschäftsbereich

Missionsverwaltung

Teilbereich

Arbeitseinteilung

- Veränderung der Datentypen CLOB
- Partitionierungen für Partition-Pruning und parallele Ausführung
- Indiziert



■ Constraints

The diagram illustrates database constraints between two tables: **PERSON** and **PERSON_TERMINE_GUELTIG**.

PERSON Table:

Column	Data Type	Constraint
id	NUMBER(38)	<pk>
name	CHAR(50 CHAR)	
geburtsdatum	DATE	
anreise_datum	DATE	
abreise_datum	DATE	
nationalität	CHAR(50 CHAR)	
gewicht	NUMBER(6,3)	
IBAN	CHAR(50 CHAR)	

Constraints:

- wurde vergeben an:** A blue arrow points from the **id** column of the **PERSON** table to the **PERSON_TERMINE_GUELTIG** table.
- hat erlangt:** A blue arrow points from the **PERSON_TERMINE_GUELTIG** table to the **anreise_datum** column of the **PERSON** table.

Table Properties - PERSON (PERSON)

General | Columns | Data Protection | Indexes | Keys | Triggers

Physical Options (Common) | Definition | Rules | Preview

Name	Code
PERSON_TERMINE_GUELTIG	PERSON_TERMINE_GUELTIG

More >> OK Cancel Apply Help

Business Rule Properties - PERSON_TERMINE_GUELTIG (PERSON_TER...

General | Expression | Definition | Preview

```
(GEBURTSDATUM < ANREISE_DATUM)
AND (
  (ABREISE_DATUM IS NULL)
  OR (ANREISE_DATUM < ABREISE_DATUM)
)
```

Server / Client

More >> OK Cancel Apply Help

■ Trigger

Diagram showing a table **RAUM** with columns:

Column	Data Type	Constraints
id	NUMBER(38)	<pk>
GEBAEUDE_standort_x	NUMBER(6)	<fk>
GEBAEUDE_standort_y	NUMBER(6)	<fk>
bezeichnung	CHAR(50 CHAR)	
raum	CHAR(10 CHAR)	
stockwerk	NUMBER(4)	
grösse	NUMBER(8,2)	
ausstattung	VARCHAR2(1000)	

Relationships:

- RAUM is a part of GEBAEUDE (ist Teil von).
- GEBAEUDE consists of RAUM (besteht aus).

Table Properties - RAUM (RAUM)

Name	Code	G
TRG_RAUM_B_INS	TRG_RAUM_B_INS	<input checked="" type="checkbox"/>
TRG_RAUM_B_UPD_RESTRICT	TRG_RAUM_B_UPD_RESTRICT	<input checked="" type="checkbox"/>

Trigger Properties - TRG_RAUM_B_INS (TRG_RAUM_B_INS)

General | Body | Template Items | Definition | Preview

BeforeInsertTrigger (From DBMS)

before

Event: insert

Ln 23, Col 56

```

create trigger %TRIGGER% before insert
on %TABLE% for each row
declare
    maxStockwerke      NUMBER(38);
    stockMax           NUMBER(38);
    stockMin           NUMBER(38);

begin
    --- Anzahl 'stockwerk' im verlinkten GEBAEUDE ermitteln
    select geb.STOCKWERKE
    into maxStockwerke
    from GEBAEUDE geb
    where (geb.STANDORT_X, geb.STANDORT_Y) = (:new.GEBAEUDE_STANDORT_X, :new.GEBAEUDE_STANDORT_Y);

    --- Minimum und Maximum von 'stockwerk' für alle RAEUME des GEBAEUDES ermitteln
    select MIN(raum.STOCKWERK), MAX(raum.STOCKWERK)
    into stockMin, stockMax
    from RAUM raum
    where (raum.GEBAEUDE_STANDORT_X, raum.GEBAEUDE_STANDORT_Y) = (:new.STANDORT_X, :new.STANDORT_Y);

    --- wird mit 'new.stockwerk' die Anzahl 'stockwerke' im GEBAEUDE überschritten ?
    if ((GREATEST(stockMax, :new.STOCKWERK - LEAST(stockMin, :new.STOCKWERK) + 1) > maxStockwerke) then
        --- in diesem Fall, mit Fehlerausgabe abbrechen
        raise_application_error(-20001, 'FEHLER: Stockwerk des neuen Raums passt nicht zum Gebäudes');
    else
        -- RAUM verwendet RAUM_SEQ als Sequenzgenerator
        select RAUM_SEQ.NEXTVAL INTO :new.ID from dual;
    end if;
end
    
```

BEISPIELDATENGENERIERUNG



- Gebäude
 - Räume
 - Lager
 - Position in Abhängigkeit von Lagergröße
- Qualifikationen
- Alle Jobs mit geforderten Qualifikationen und Arbeitszeiten
 - Via Schleife für die ersten 12 Wochen erstellt
- Personen für jeden Job erstellen
 - Mit Nachweis über entsprechende Qualifikation
 - Bewerbungen hinzufügen
 - Vergaben der Jobs hinzufügen
- Posten erzeugen
 - Geräte und Ressourcen erzeugen
- Lieferanten erstellen
 - Angebote erstellen

BEISPIELDATENGENERIERUNG



```
--- Personen hinzufügen ---
declare
    per_geburtsdatum    DATE := to_date('1970-01-01', 'YYYY-MM-DD');
    per_gewicht         NUMBER(6,3) := 65.230;

begin
    --- etwa 300 PERSONEN aus diversen Ländern erzeugen
    for nname in (select * from TMP_NACHNAMEN) loop
        for vname in (select * from TMP_VORNAMEN where nation = nname.nation) loop
            insert into PERSON(name, geburtsdatum, anreise_datum, nationalitat, gewicht)
            values(
                (rtrim(vname.name) || ' ' || rtrim(nname.name)),
                per_geburtsdatum,
                to_date('2022-01-04', 'YYYY-MM-DD'),
                vname.nation,
                per_gewicht
            );

            per_geburtsdatum := per_geburtsdatum + 11;
            per_gewicht := per_gewicht + 0.100;
        end loop;
    end loop;
end;
/
```

```
--- Qualifikationen für Personen hinzufügen ---
declare
    personid            NUMBER(38);
    persongeburtsdatum  DATE;
    nachweisid          NUMBER(38);
    vergabeid           NUMBER(38);
    fakultatname        CHAR(50 CHAR);
    nachweisnote        NUMBER(1) := 1;

begin
    --- ID der ersten Person aus der Liste holen
    select min(id)
    into personid
    from PERSON;

    --- über alle JOBS iterieren
    for job in (select * from JOB order by id asc) loop
        --- alle Anforderungen des JOBS durchgehen
        for jobquali in (select * from JOBQUALIFIKATION where JOB_id = job.id) loop
            select geburtsdatum
            into persongeburtsdatum
            from PERSON
            where id = personid;

            --- NACHWEIS für diese für diese Person, für diese Anforderung hinzufügen
            insert into NACHWEIS(PERSON_id, datum, fakultat, note)
            values(
                personid,
                persongeburtsdatum + (25 * 365),
                'unbekannt',
                nachweisnote
            );
            select NACHWEIS_seq.currval into nachweisid from dual;

            --- QUALIFIKATIONSNACHWEIS für diese für diese Person, für diese Anforderung hinzufügen
            insert into QUALIFIKATIONSNACHWEIS(NACHWEIS_id, QUALI_bezeichnung)
            values(nachweisid, jobquali.QUALI_bezeichnung);
        end loop;
    end loop;
```

LIVE SQL DEMO



AUSBLICK – FINALE ABGABE



- Auf Feedback eingehen
- Semantische Modellierung
- Dokumentation
- Projektbericht
- Alle Projektdaten hochladen



**Lunar Prospectors bedankt sich
für Ihre Aufmerksamkeit !**