

FRAGEN ZUR PROBEKLAUSUR

WIEDERHOLUNG

```
CREATE VIEW BOHRMASCHINEN (NR, EINKAUFSWERT, AKTUELLER_WERT)
AS
SELECT MNR, NEUWERT, ZEITWERT
FROM MASCHINE
WHERE NAME='Bohrmaschine';

SELECT *
FROM BOHRMASCHINEN WHERE AKTUELLER_WERT > 16000;
```

AUFGABEN

- 1. ERSTELLEN SIE EINE VIEW, DIE NUR MASCHINEN ENTHALT, DEREN ZEITWERT WENIGER ALS 60% DES NEUWERTS IST**
- 2. ERSTELLEN SIE EINE VIEW MIT MITARBEITERN, DIE MEHR ALS 3000.- VERDIENEN**

LOS GEHT'S

\o/

INTEGRITÄT

INTEGRITÄT UND ACID

SEMANTISCHE INTEGRITÄT C (CONSISTENCY, KONSISTENZ)

PHYSISCHES INTEGRITÄT

D (DAUERHAFTIGKEIT)

A (ATOMARITÄT)

ABLAUFINTEGRITÄT I (ISOLATION)

INTEGRITÄTSBEDINGUNGEN

BEISPIELE

**UBERZIEHUNGEN DER GIROKONTEN VON
STUDENTEN SIND NUR BIS 10.000 EUR
GESTATTET**

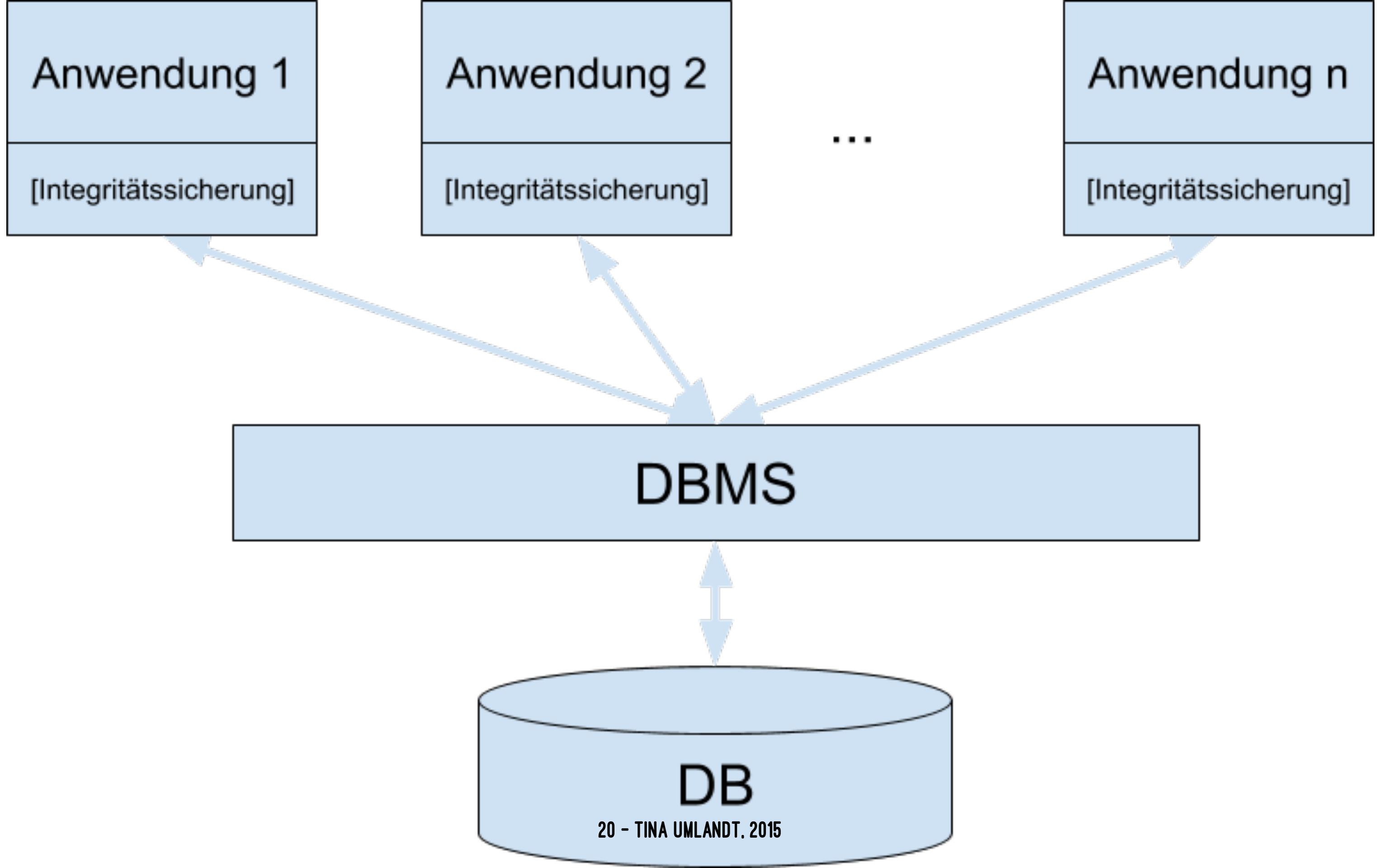
**DIE SUMME ALLER KONTEN IST STETS
NULL**

**ANDERUNGEN VON KONTEN SIND PRO
TRANSAKTION AUF 10 MIO. BEGRENZT**

INTEGRITÄTSSICHERUNGSKOMPONENTE

MÖGLICHKEITEN

1. PROGRAMMCODE



VORTEILE

FLEXIBILITÄT DER PROGRAMMIERSPRACHE

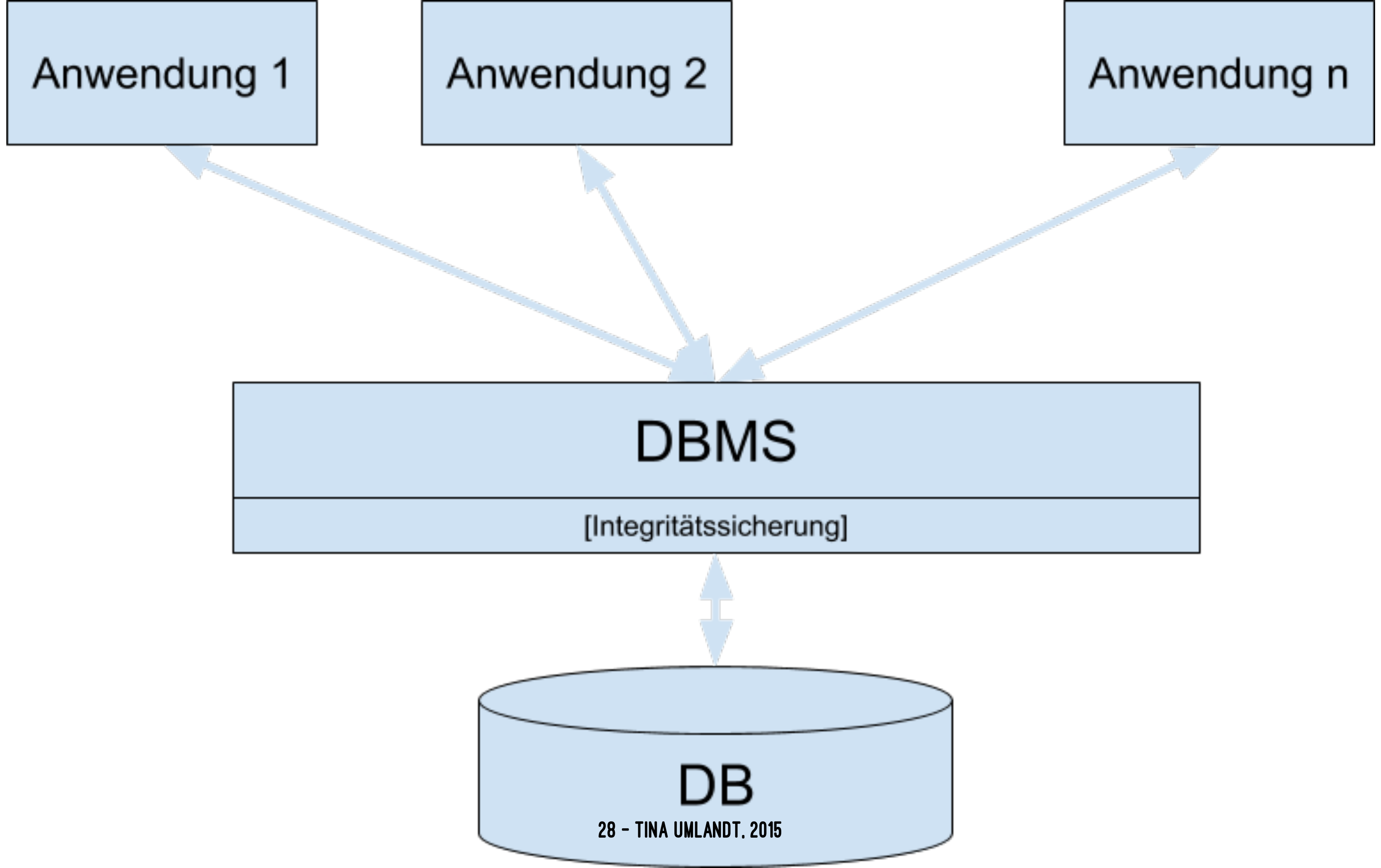
UNABHÄNGIG VOM DBMS

NACHTEILE

KEINE ZENTRALE KONTROLLE

REDUNDANZ

2. DBMS



VORTEILE

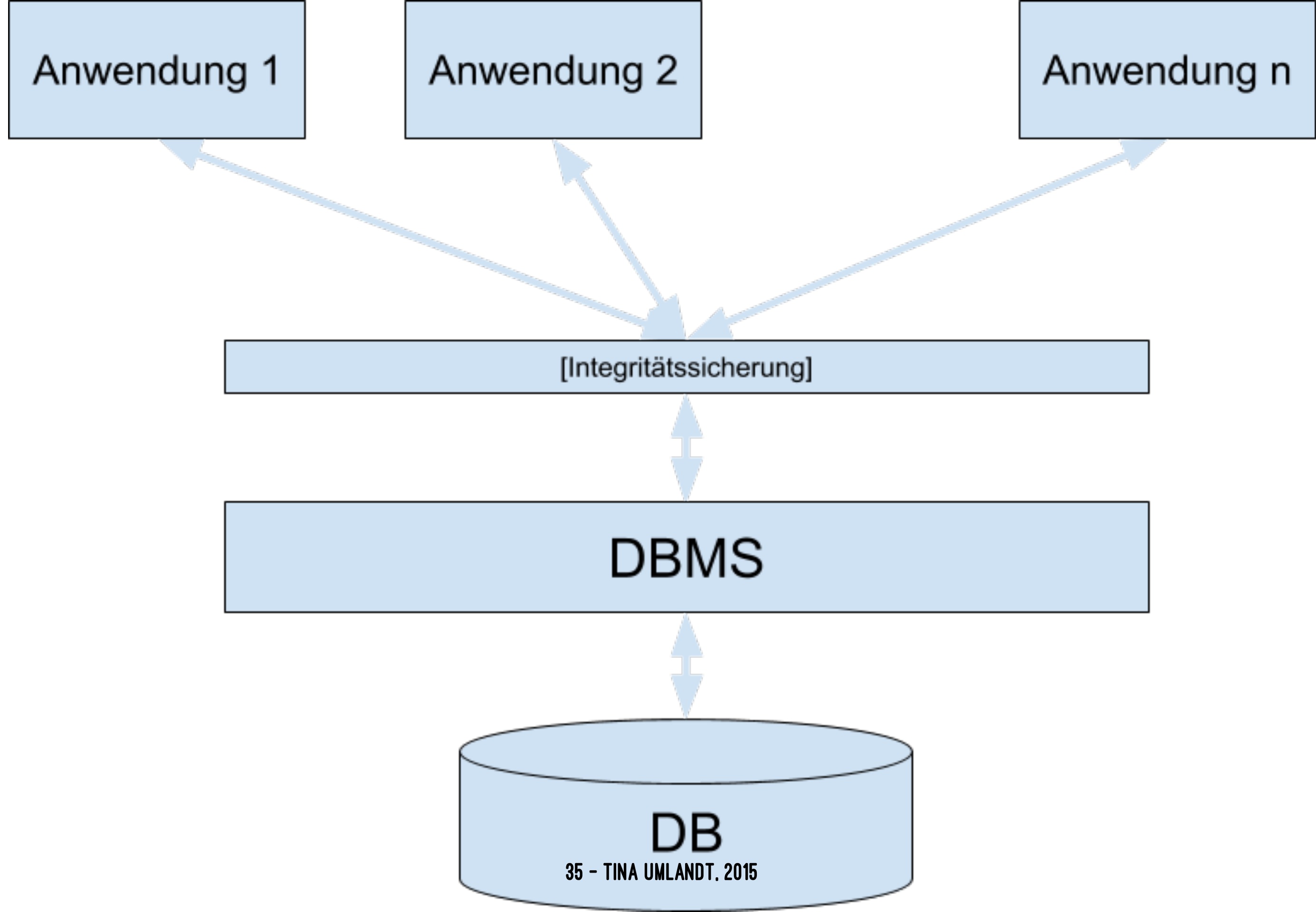
ANWENDUNGSUNABHÄNGIG

REDUNDANZFREI

NACHTEIL

ABHÄNGIG VON SPRACHMITTELN DES DBMS

3. SCHNITTSTELLE



VORTEILE

ANWENDUNGSUNABHÄNGIG

REDUNDANZFREI

UNABHÄNGIG VOM DBMS

NACHTEIL

**GEFAHR, DASS AN KAPSELUNG
„VORBEIPROGRAMMIERT“ WIRD**

KLASSIFIKATION

> GRANULARITÄT

➤ **AUSDRUCKSFÄHIGKEIT DER SPRACHE**

> ANZAHL DER BETRACHTETEN DATENBANKZUSTÄNDE

> ZEITPUNKT DER ÜBERPRÜFUNG

> REAKTION

BEISPIEL FÜR KLASSIFIKATION VON INTEGRITÄTSBEDINGUNGEN

UBERZIEHUNGEN DER GIROKONTEN VON STUDENTEN SIND NUR BIS
10.000 EUR GESTATTET

GRANULARITÄT: TUPEL
AUSDRUCKSFÄHIGKEIT: SQL
EIN DB-ZUSTAND
ZEITPUNKT: IMMER
REAKTION: ABBRECHEN

DIE SUMME ALLER KONTEN IST STETS NULL.

GRANULARITÄT: RELATION
AUSDRUCKSFÄHIGKEIT: SQL
EIN DB-ZUSTAND
ZEITPUNKT: TRANSAKTIONSENDE
REAKTION: ABBRECHEN

ANDERUNGEN VON KONTEN SIND PRO TRANSAKTION AUF 10 MIO.
BEGRENZT

GRANULARITÄT: TUPEL
AUSDRUCKSFÄHIGKEIT: SQL
ZWEI DB-ZUSTÄNDE
ZEITPUNKT: TRANSAKTIONSENDE
REAKTION: ABBRECHEN

FORMULIERUNG VON INTEGRITÄTSBEDINGUNGEN

INHÄRENTE INTEGRITÄTSBEDINGUNGEN

➤ **TYPINTEGRITÄT**

➤ **SCHLÜSSELINTEGRITÄT**

➤ REFERENTIELLE INTEGRITÄT

EXPLIZITE INTEGRITÄTSBEDINGUNGEN

➤ NULL / NOT NULL

➤ PRIMARY KEY

› FOREIGN KEY (<Attribute>)
REFERENCES <tabelle>
(<Attribute>)

› CHECK (Suchbedingung)

```
CREATE TABLE divisions
  (div_no      NUMBER      CONSTRAINT check_divno
    CHECK (div_no BETWEEN 10 AND 99)
    DISABLE,
  div_name    VARCHAR2(9)  CONSTRAINT check_divname
    CHECK (div_name = UPPER(div_name))
    DISABLE,
  office      VARCHAR2(10) CONSTRAINT check_office
    CHECK (office IN ('DALLAS', 'BOSTON',
      'PARIS', 'TOKYO'))
    DISABLE);
```

ORACLE¹

¹ [HTTPS://DOCS.ORACLE.COM/CD/B19306_01/SERVER.102/B14200/CLAUSES002.HTM#11002719](https://docs.oracle.com/cd/B19306_01/SERVER.102/B14200/CLAUSES002.HTM#11002719)

TRIGGER

```
CREATE OR REPLACE TRIGGER Print_maschine_changes
  BEFORE DELETE OR INSERT OR UPDATE ON maschine
  FOR EACH ROW
  WHEN (new.ZEITWERT > 0)
  DECLARE
    zeitwert_diff number;
  BEGIN
    zeitwert_diff := :new.ZEITWERT - :old.ZEITWERT;
    dbms_output.put_line('Alter Zeitwert: ' || :old.ZEITWERT);
    dbms_output.put_line('Neuer Zeitwert: ' || :new.ZEITWERT);
    dbms_output.put_line(' -> Differenz: ' || zeitwert_diff);
  END;
/
```


EREIGNISSE

INSERT, UPDATE, DELETE

CREATE, ALTER, DROP

```
CREATE OR REPLACE TRIGGER drop_trigger
BEFORE DROP ON KIND
BEGIN
    RAISE_APPLICATION_ERROR (
        num => -20000,
        msg => 'Cannot drop KIND table!!11elf');
END;
/
```

DATENBANKINTERNE EREIGNISSE

ZEITPUNKTE

before / after

instead of

EIGENSCHAFTEN

referencing (new|old) as

LEVEL

for each (statement | row)

```
CREATE TRIGGER gehalt_neu_upd_bef  
BEFORE update on GEHALT  
for each row  
when (new.BETRAG / old.BETRAG > 1.1 ) BEGIN  
    insert into GEHALT_ALT  
    values (:old.GEH_STUFE, :old.BETRAG);  
END;
```

REIHENFOLGE

1. AUSFÜHRUNG VON BEFORE - ANWEISUNGSTRIGGER

2 . AUSFÜHRUNG VON BEFORE - ZEILENTRIGGER

3 . AUSFUHRUNG DES EIGENTLICHEN SQL-STATEMENTS

4 . ABARBEITUNG DER AFTER - ZEILENTRIGGER FÜR JEDE ZEILE

DIE PUNKTE 2. BIS 4. WIEDERHOLEN SICH JETZT SO OFT
WIE ZEILEN BETROFFEN SIND

5 . AUSFÜHRUNG DER AFTER - ANWEISUNGSTRIGGER

SYNTAX

```
CREATE [OR REPLACE] TRIGGER [user.]triggername  
{BEFORE | AFTER | INSTEAD OF}  
{INSERT | UPDATE [OF column [, column] ... ] | DELETE}  
[OR {INSERT | UPDATE [OF column [, column] ... ] | DELETE} ]  
ON [user. ] {TABLE | VIEW}  
[FOR EACH {ROW | STATEMENT}]  
[WHEN Bedingung]  
Anweisungsblock
```

```
CREATE TRIGGER gehalt_upd_ins
BEFORE insert OR update on GEHALT
for each row
BEGIN
    IF INSERTING THEN
        insert into GEHALT_ALT
        values (:new.GEH_STUFE, :new.BETRAG);
    ELSE -- not inserting, then we are updating BETRAG
        insert into GEHALT_ALT
        values (:old.GEH_STUFE, :old.BETRAG);
    END IF;
END;
/
```

MEHR BEISPIELE


```
CREATE TABLE verlauf (  
datum    date ,  
pnr number(4),  
pnr_neu number(4)  
);
```

```
CREATE OR REPLACE TRIGGER verlauf_tr  
BEFORE UPDATE ON personal  
FOR EACH ROW  
BEGIN  
    INSERT INTO verlauf (datum, pnr, pnr_neu)  
    VALUES (sysdate, :old.pnr, :new.pnr);  
END;  
/
```

```
SELECT *  
FROM verlauf;
```

```
UPDATE PERSONAL  
SET PNR=500 WHERE PNR=177;
```

```
SELECT *  
FROM verlauf;
```

```
ALTER TABLE personal ADD COLUMN geh_betrag int;  
UPDATE PERSONAL SET geh_betrag = (  
    SELECT betrag FROM gehalt WHERE gehalt.geh_stufe=PERSONAL.geh_stufe  
);
```

```
CREATE OR REPLACE TRIGGER peronal_gehalt  
AFTER UPDATE OF geh_stufe ON personal  
FOR EACH ROW  
BEGIN  
    UPDATE PERSONAL SET geh_betrag =  
        (SELECT betrag FROM gehalt WHERE geh_stufe=:new.geh_stufe);  
END;  
/
```

AUFGABEN

- 1. SCHREIBEN SIE EINEN TRIGGER, DER BEI EINER ANDERUNG DES ZEITWERTS DER MASCHINEN DEN VERLUST IN PROZENT ANGIBT**
- 2. WENN EINEM MITARBEITER EIN KIND HINZUGEFÜGT WIRD, SOLL SIE AUTOMATISCH EINE PRÄMIE IN HÖHE VON 300.- BEKOMMEN.**
- 3. ALLE UPDATES DER ZEITWERTE DER TABELLE MASCHINE SOLLEN IN EINER NEUEN TABELLE MASCHINE_LOG PROTOKOLIIERT WERDEN: ALTER UND NEUER WERT UND DATUM**

4.

5.



**DAS WAR'S FÜR
HEUTE**