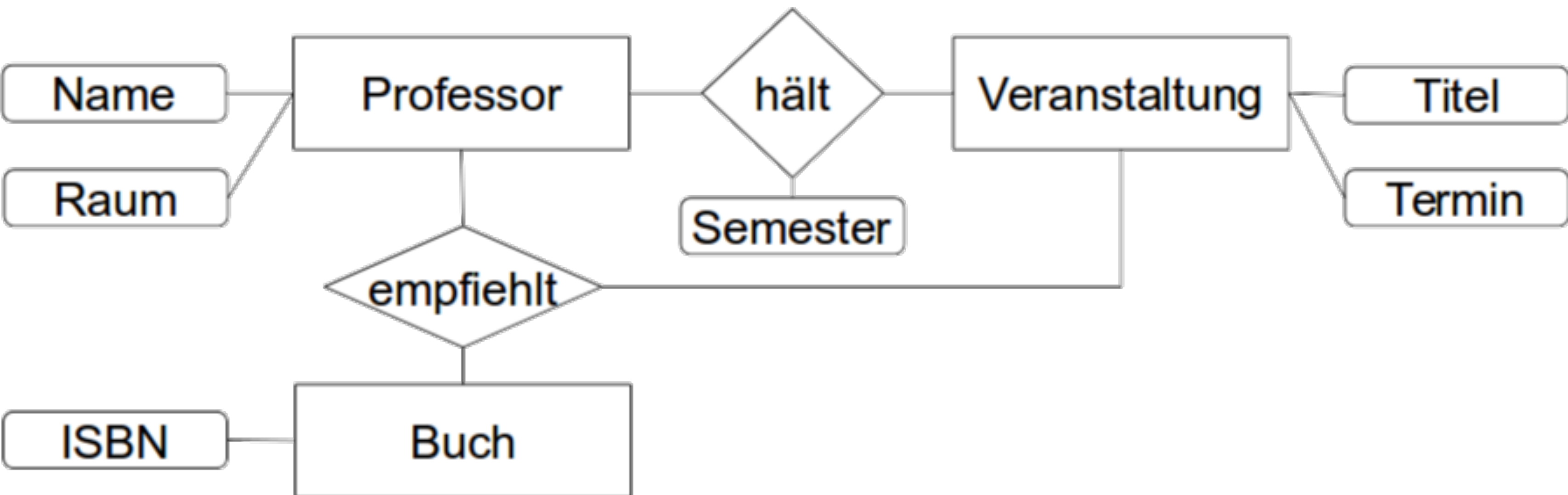


NÄCHSTES MAL: ORACLE

WIEDERHOLUNG

WARUM GIBT ES VERSCHIEDENE DATENMODELLE?

ENTITY- RELATIONSHIP- MODELL

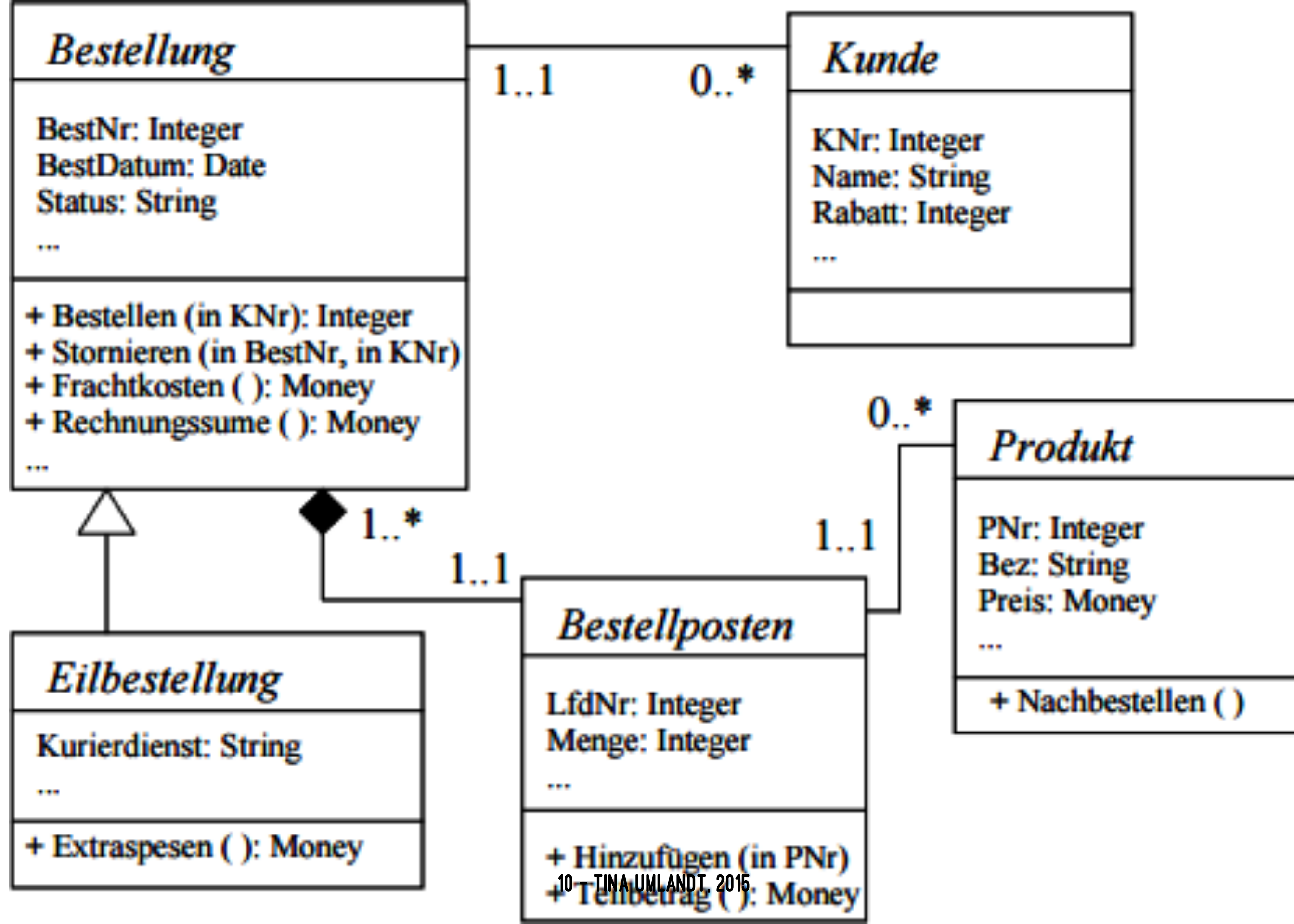


AUFGABE:

**SIE SOLLEN EINE TODO-APP MIT EINEM ER-MODELL MODELLIEREN.
FOLGENDE REGELN:**

- **TODO HAT IMMER TEXT**
- **KANN MEHRERE BILDER ENTHALTEN**
- **HAT MINDESTENS EINEN AUTOR**
- **MAN MUSS HERAUSFINDEN KOENNEN, WANN DIE TODOS ERSTELLT
UND GEAENDERT WURDEN**

UML

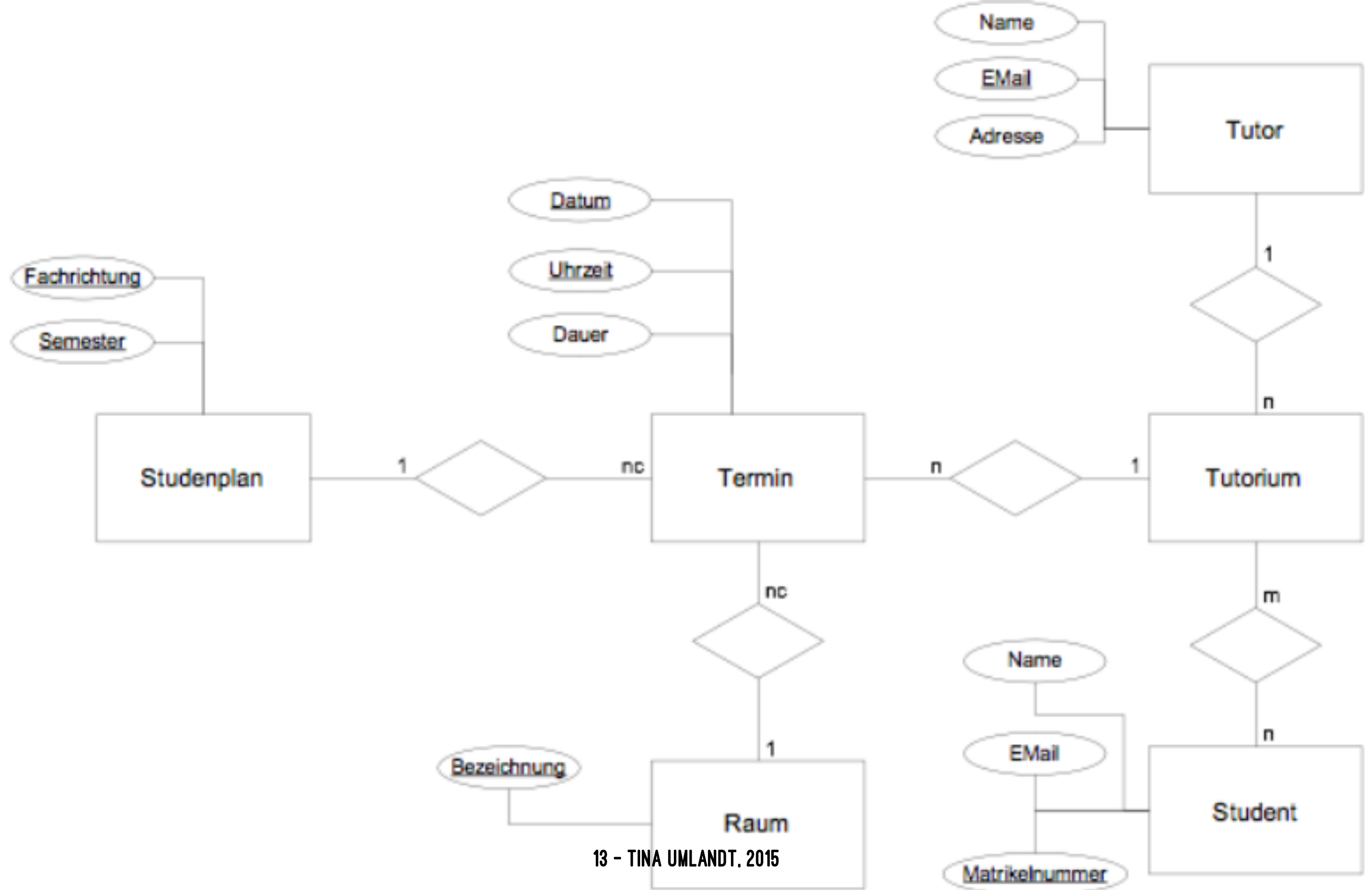


RELATIONENSCHEMATA

TRANSFORMATION:

ERM → RELATIONENSHEMA

- **JEDER ENTITÄTSTYP WIRD AUF EINE RELATION ABGEBILDET**
- **DIE ATTRIBUTE DER ENTITÄTSTYPEN WERDEN DEN ATTRIBUTEN DER RELATION ZUGEORDNET**
 - **ES MUSS EIN PRIMÄRSCHLÜSSEL GEWÄHLT WERDEN**
 - **ABSCHLIEßEND WERDEN DIE BEZIEHUNGEN TRANSFORMIERT**
- **BEZIEHUNGSTYP INNERHALB DES ERM WIRD MITTELS EINES EIGENEN FREMDSCHLÜSSELS IM RELATIONENMODELL ABGEBILDET**
- **NM BZW. N:MC – BEZIEHUNGEN WERDEN IN ZWEI 1:N BZW. 1:NC – BEZIEHUNGEN AUFGELÖST.**



SCHLÜSSEL

SUPERSCHLÜSSEL

SCHLÜSSELKANDIDAT

AUFGABE: BESTIMMEN DER SUPERSCHLUSSEL

-----	-----	-----
Name	Nummer	Königshaus
-----	-----	-----
Edward	II	Plantagenet
-----	-----	-----
Edward	III	Plantagenet
-----	-----	-----
Richard	III	Plantagenet
-----	-----	-----
Henry	IV	Lancaster
-----	-----	-----

PRIMÄRSCHLÜSSEL

Buch

BuchId	ISBN	Titel
1	4FA33	Hamlet
2	E5B11	Die Verwandlung

FREMSCHLÜSSEL

BuchId	ISBN	Titel
1	4FA33	Hamlet
2	E5B11	Die Verwandlung
3	CD652	King Lear

AutorId	Vorname	Nachname
1	William	Shakespeare
2	Franz	Kafka

BuchId	AutorId
1	1
2	2
2	1

WIEDERHOLUNG DER DATENMODELLE

HIERARCHISCHES MODELL

NETZWERKMODELL

OBJEKT-ORIENTIERTES MODELL

OBJEKT-RELATIONALES MODELL

XML-BASIERENDES MODELL

LOS GEHT'S

\o/

NOSQL- DATENMODELLE

KEY-VALUE-STORE

```
shop.settings.vat=19
```

```
shop.country="de_DE"
```

DOKUMENTENDATENBANKEN

BEISPIEL: COUCHDB-DOKUMENT

```
{
  "_id" : "00a271787f89c0ef2e10e88a0c0001f4"
  "_rev": "5509377776",
  "name": "Peter Lustig",
  "address": {
    "street": "Teststr.",
    "city": "Hamburg"
  }
}
```

ZUSAMMENFASSUNG

RELATIONALES DATENBANK-DESIGN

SCHREIBWEISEN

```
Autor = (Autorvorname, Autornachname, Verlag,  
Buchtitel, ISBN, Preis, Sprache, Seitenanzahl)
```

```
Autor = (  
    Autorvorname,  
    Autornachname,  
    Verlag,  
    Buchtitel,  
    ISBN,  
    Preis,  
    Sprache,  
    Seitenanzahl  
)
```

EINE TABELLE FUER ALLES

Autorvorname	Autornachname	Verlag	Buchtitel	ISBN	Preis	Sprache	Seitenanzahl
Franz	Kafka	Ideenbrücke Verlag	Die Verwandlung	978-3-945909-67-6	3,99 EUR	Deutsch	64
Franz	Kafka	Philipp Reclam Jun.	Die Verwandlung	978-3-15-009900-1	2,00 EUR	Deutsch	80
Franz	Kafka	Fischer Taschenbuch Verlag	Das Urteil	978-3-596-20019-1	6,95 EUR	Deutsch	185

FUNKTIONALE ABHÄNGIGKEITEN

(FUNCTIONAL DEPENDENCY – FD)

DEFINITION:

Sei V eine Attributmenge, $X, Y \subseteq V$ und $r \in \text{Rel}(V)$

Dann ist Y funktional abhängig von X ,
notiert als $X \rightarrow Y$, wenn und nur wenn jeder
 X -Wert von r genau einen eindeutigen Y -Wert
in r bestimmt.

A	B	C	D	E
1	1	1	1	1
1	0	1	1	1
2	2	0	0	1
2	3	2	0	1

MENGE F VON FDS:

$$F = \{AB \rightarrow C, A \rightarrow D, D \rightarrow E\}$$

TEST, OB EINE RELATION $r \in \text{Rel}(V)$ EINE FD DER FORM $X \rightarrow Y$ ERFÜLLT:

- SORTIEREN VON R NACH WERTEN DES ATTRIBUTES X.**
- PRÜFEN ALLER TUPEL MIT GLEICHEN X-WERTEN AUF GLEICHE Y-WERTE.**

Prof	Veranstaltung	Student	Buchtitel	Autor	ISBN	Semester
Rainbow	Ponylehre	Clara Clayton	Meine Ponies	Spike	4711	3
Rainbow	Ponylehre	Buffy	Meine Ponies	Spike	4711	2
Dr. Brown	Future 101	Clara Clayton	In Hill Valley	Marty McFly, Jennifer Parker	4713	3

Prof	Veranstaltung	Student	Buchtitel	Autor	ISBN	Semester
Rainbow	Ponylehre	Clara Clayton	Meine Ponies	Spike	4711	3
Rainbow	Ponylehre	Buffy	Meine Ponies	Spike	4711	2
Dr. Brown	Future 101	Clara Clayton	In Hill Valley	Marty McFly, Jennifer Parker	4713	3

TRIVIALE FUNKTIONALE ABHÄNGIGKEIT

$$X \rightarrow X$$

VOLLE FUNKTIONALE ABHÄNGIGKEIT

$X = \{X_1, X_2, \dots, X_n\}$

$Y = \{Y_1, Y_2, \dots, Y_m\}$

$X \rightarrow Y$

Y ist voll funktional abhängig von X , wenn
 Y funktional abhängig von X , aber nicht
funktional abhängig von einer echten
Teilmenge von X ist.

EIGENSCHAFTEN DER FUNKTIONALEN ABHÄNGIGKEITEN

**SCHLUSSEL SIND SPEZIALFALLE VON
FUNKTIONALEN ABHÄNGIGKEITEN.**

EIN SCHLUSSEL X LIEGT VOR, WENN
FÜR EIN RELATIONENSHEMA R EINE
FUNKTIONALE ABHÄNGIGKEIT $X \rightarrow Y$
GILT UND X MINIMAL IST.

**FUNKTIONALE ABHÄNGIGKEITEN SIND
AUSSAGEN ÜBER DAS SCHEMA, NICHT
ÜBER EINE AUSPRÄGUNG**

**WENN MAN EINIGE FD GEFUNDEN HAT.
DANN GIBT ES EINEN ALGORITHMUS, UM
WEITERE FD DARAUS ABZULEITEN**

AUS EINER MENGE VON GEGEBENEN FD
(ALS F BEZEICHNET) KÖNNEN MEIST
WEITERE FD ABGELEITET WERDEN

F +

TRANSITIVE HÜLLE

ARMSTRONGSCHE AXIOME

REFLEXIVITÄT

WENN $Y \subseteq X$, DANN $X \rightarrow Y$ IN F^+

(X, Y, Z SIND MENGEN VON ATTRIBUTEN):

ERWEITERBARKEIT

**WENN $X \rightarrow Y$ IN F^+ , DANN $XZ \rightarrow YZ$ IN F^+ FÜR BELIEBIGE Z
(X, Y, Z SIND MENGEN VON ATTRIBUTEN):**

TRANSITIVITÄT

WENN $X \rightarrow Y$ UND $Y \rightarrow Z$ IN F^+ , DANN $X \rightarrow Z$ IN F^+

(X, Y, Z SIND MENGEN VON ATTRIBUTEN):

ANOMALIEN

EINFÜGEANOMALIE

LÖSCHANOMALIE

ÄNDERUNGSANOMALIE

NORMALFORMEN

Prof	Veranstaltung	Student	Buchtitel	Autor	ISBN	Semester
Rainbow	Ponylehre	Clara Clayton	Meine Ponies	Spike	4711	3
Rainbow	Ponylehre	Buffy	Meine Ponies	Spike	4711	2
Dr. Brown	Future 101	Clara Clayton	In Hill Valley	Marty McFly, Jennifer Parker	4713	3

ERSTE NORMALFORM

1NF

1. ALLE WERTE ENTSTAMMEN AUSSCHLIEßLICH ATOMAREN WERTEBEREICHEN

Inventar- nummer	Buchtitel	Autor	ISBN	Lagerbestand
001	Meine Ponies	Spike	4711	1
002	In Hill Valley	Jennifer Parker, Marty McFly	4713	4
003	PostgreSQL Administration	Peter Eisentraut, Bernd Helmle	6565	3

Inventar- nummer	Buchtitel	Autor	ISBN	Lagerbestand
001	Meine Ponies	Spike	4711	1
002	In Hill Valley	Jennifer Parker	4713	4
002	In Hill Valley	Marty McFly	4713	4
003	PostgreSQL Administration	Peter Eisentraut	6565	3
003	PostgreSQL Administration	Bernd Helmle	6565	3

Inventar- nummer	Buchtitel	Autorvorname	Autornachname	ISBN	Lagerbestand
001	Meine Ponies	Spike	Spike	4711	1
002	In Hill Valley	Jennifer	Parker	4713	4
002	In Hill Valley	Marty	McFly	4713	4
003	PostgreSQL Administration	Peter	Eisentraut	6565	3
003	PostgreSQL Administration	Bernd	Helmle	6565	3

ZWEITE NORMALFORM

2NF

1. RELATION IST IN 1NF

**2. KEIN NICHT-SCHLÜSSELATTRIBUT IST FUNKTIONAL ABHÄNGIG
VON EINER ECHTEN TEILMENGE EINES SCHLÜSSELKANDIDATEN**

Inventar- nummer	Buchtitel	Autorvorname	Autornachname	ISBN
001	Meine Ponies	Spike	Sparkle	4711
002	In Hill Valley	Jennifer	Parker	4713
002	In Hill Valley	Marty	McFly	4713
003	PostgreSQL Administration	Peter	Eisentraut	6565
003	PostgreSQL Administration	Bernd	Helmle	6565

Buch

Inventar-nummer	Buchtitel	ISBN
001	Meine Ponies	4711
002	In Hill Valley	4713
003	PostgreSQL Administration	6565

Buchautor

Inventar-nummer	Autorvorname	Autornachname
001	Spike	Sparkle
002	Jennifer	Parker
002	Marty	McFly
003	Peter	Eisentraut
003	Bernd	Helmle

DEKOMPOSITION

SEI

$R = (\text{Schlüsselteil1}, \text{Schlüsselteil2}, \text{NichtSchlüssel1}, \text{NichtSchlüssel2})$
 $FD = \{\text{Schlüsselteil2} \rightarrow \text{NichtSchlüssel2}\}$

DANN GILT

$R1 = (\text{Schlüsselteil1}, \text{Schlüsselteil2}, \text{NichtSchlüssel1})$
 $R2 = (\text{Schlüsselteil2}, \text{NichtSchlüssel2})$

Name	Wohnort	PLZ
Spike	Hamburg	22761
McFly	Hamburg	22761
Helmle	Rostock	23456

Name -> {Wohnort, PLZ}

PLZ -> Wohnort

Name -> PLZ -> Wohnort

Name	Wohnort	PLZ
Spike	Hamburg	22761
McFly	Hamburg	22761
Helmle	Rostock	23456

DRITTE NORMALFORM

3NF

1. RELATION IST IN 2NF

**2. JEDES ATTRIBUT AUßERHALB DES PRIMÄRSCHLÜSSELS IST
DIREKT ABHÄNGIG VOM PRIMÄRSCHLÜSSEL (INSBESONDERE
KEINE TRANSITIVEN ABHÄNGIGKEITEN)**

Name	PLZ
Spike	22761
McFly	22761
Helmle	23456

Wohnort	PLZ
Hamburg	22761
Rostock	23456

DEKOMPOSITION

SEI

$R = (\text{Schlüsselteil1}, \text{NichtSchlüssel1}, \text{NichtSchlüssel2})$
 $FD = \{\text{NichtSchlüssel1} \rightarrow \text{NichtSchlüssel2}\}$

DANN ERGIBT SICH

$R1 = (\text{Schlüsselteil1}, \text{NichtSchlüssel1})$
 $R2 = (\text{NichtSchlüssel1}, \text{NichtSchlüssel2})$

BOYCE-CODD-NORMALFORM

BCNF

VIERTE / FÜNFTE NORMALFORM

4NF / 5NF

AUFGABE

Vorname	Nachname	Geburtstag	Einstellungsdatum	Abteilungsname	Abteilungsleiter	Projektname	Projektstart	Projektverantwortliche Abteilung
Ada	Lovelace	10.12.65	01.06.90	Informatik	Lovelace	Möbius Umzug nach Berlin	12.14 10.13	Informatik Controlling
Grace	Hopper	21.12.60	01.06.87	Informatik	Lovelace	Möbius	12.14	Informatik
Jean	Bartik	09.03.61	01.03.84	Finanzen	Euler	Umzug nach Berlin	10.13	Controlling
Howard	Aiken	22.07.69	01.09.98	Informatik	Lovelace	Möbius	12.14	Informatik
Pierre	de Fermat	30.01.55	01.01.72	Controlling	Descartes	Umzug nach Berlin	10.13	Controlling
René	Descartes	09.10.50	01.01.90	Controlling	Descartes	Umzug nach Berlin Möbius	10.13 12.14	Controlling Informatik
Leonhard	Euler	16.04.68	01.09.97	Finanzen	Euler	Umzug nach Berlin Integrale	10.13 04.15	Controlling Informatik
John	Wilson	03.09.73	01.09.99	Informatik	Lovelace	Umzug nach Berlin Integrale	10.13 04.15	Controlling Informatik

DATENBANKENTWURF

KRITERIEN

- 1. REDUNDANZFREIE INFORMATIONEN**
- 2. VOLLSTÄNDIGE DARSTELLUNG ALLER DATEN**
- 3. KONSISTENZ DES ENTWURFS**
- 4. SPRACHE IST AUSDRUCKSSTARK UND VERSTÄNDLICH**
- 5. BESCHREIBUNGSKONSTRUKTE SIND KLAR DEFINIERT**
- 6. DOKUMENTE SIND LEICHT LESBAR**

MODELLIERUNG EINER SCHULE ?!

DATENDEFINITION

DATA DEFINITION LANGUAGE

DDL

CREATE TABLE
ALTER TABLE
DROP TABLE

CREATE VIEW

DROP VIEW

BZW ORACLE

CREATE INDEX
ALTER INDEX
DROP INDEX

ORACLE [HTTPS://DOCS.ORACLE.COM/CD/B2835901/SERVER.111/B28286/STATEMENTS7002.HTM](https://docs.oracle.com/cd/B2835901/server.111/b28286/statements7002.htm)

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
[(create_definition,...)]
[table_options]
[partition_options]

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
[(create_definition,...)]
[table_options]
[partition_options]
select_statement

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
( LIKE old_tbl_name | (LIKE old_tbl_name) )

create_definition:
col_name column_definition
| [CONSTRAINT [symbol]] PRIMARY KEY (index_type) (index_col_name,...)
  [(index_option) ...]
| [INDEX|KEY] (index_name) (index_type) (index_col_name,...)
  [(index_option) ...]
| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
  (index_name) (index_type) (index_col_name,...)
  [(index_option) ...]
| [FULLTEXT[SPATIAL]] [INDEX|KEY] (index_name) (index_col_name,...)
  [(index_option) ...]
| [CONSTRAINT [symbol]] FOREIGN KEY
  (index_name) (index_col_name,...) reference_definition
| CHECK (expr)

column_definition:
data_type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
[COMMENT 'string']
[COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
[STORAGE {DISK|MEMORY|DEFAULT}]
[reference_definition]

data_type:
BIT[(length)]
| TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE
| TIME
| TIMESTAMP
| DATETIME
| YEAR
| CHAR[(length)] [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| VARCHAR(length) [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| BINARY[(length)]
| VARBINARY(length)
| TINYBLOB
| BLOB
| MEDIUMBLOB
| LONGBLOB
| TINYTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| TEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| MEDIUMTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| LONGTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| ENUM(value1,value2,value3,...)
  [CHARACTER SET charset_name] [COLLATE collation_name]
| SET(value1,value2,value3,...)
  [CHARACTER SET charset_name] [COLLATE collation_name]
| spatial_type

index_col_name:
col_name [(length)] [ASC | DESC]

index_type:
USING {BTREE | HASH}

index_option:
KEY_BLOCK_SIZE [=] value
| index_type
| WITH PARSER parser_name

reference_definition:
REFERENCES tbl_name (index_col_name,...)
[MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
[ON DELETE reference_option]
[ON UPDATE reference_option]

reference_option:
RESTRICT | CASCADE | SET NULL | NO ACTION

table_options:
table_option [(, table_option) ...]

table_option:
ENGINE [=] engine_name
| AUTO_INCREMENT [=] value
| AVG_ROW_LENGTH [=] value
| [DEFAULT] CHARACTER SET [=] charset_name
| CHECKSUM [=] {0 | 1}
| [DEFAULT] COLLATE [=] collation_name
| COMMENT [=] 'string'
| CONNECTION [=] 'connect_string'
| DATA DIRECTORY [=] 'absolute path to directory'
| DELAY_KEY_WRITE [=] {0 | 1}
| INDEX DIRECTORY [=] 'absolute path to directory'
| INSERT_METHOD [=] { NO | FIRST | LAST }
| KEY_BLOCK_SIZE [=] value
| MAX_ROWS [=] value
| MIN_ROWS [=] value
| PACK_KEYS [=] {0 | 1 | DEFAULT}
| PERSISTENT [=] 'string'
| ROW_FORMAT [=] (DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT)
| TABLESPACE tablespace_name [STORAGE {DISK|MEMORY|DEFAULT}]
| UNION [=] (tbl_name[,tbl_name]...)

partition_options:
PARTITION BY
  ( [LINEAR] HASH(expr)
  | [LINEAR] KEY(column_list)
  | RANGE(expr)
  | LIST(expr) )
[PARTITIONS num]
[SUBPARTITION BY
  ( [LINEAR] HASH(expr)
  | [LINEAR] KEY(column_list) )
| SUBPARTITIONS num]
)
[(partition_definition [, partition_definition] ...)]

partition_definition:
PARTITION partition_name
VALUES
  (LESS THAN ((expr) | MAXVALUE)
  |
  IN (value_list))
[([STORAGE] ENGINE [=] engine_name)
 (COMMENT [=] 'comment_text' )
 (DATA DIRECTORY [=] 'data_dir')
 (INDEX DIRECTORY [=] 'index_dir')
 (MAX_ROWS [=] max_number_of_rows)
 (MIN_ROWS [=] min_number_of_rows)
 (TABLESPACE [=] tablespace_name)
 (NODEGROUP [=] node_group_id)
 [(subpartition_definition [, subpartition_definition] ...)]

subpartition_definition:
SUBPARTITION logical_name
([STORAGE] ENGINE [=] engine_name)
 (COMMENT [=] 'comment_text' )
 (DATA DIRECTORY [=] 'data_dir')
 (INDEX DIRECTORY [=] 'index_dir')
 (MAX_ROWS [=] max_number_of_rows)
 (MIN_ROWS [=] min_number_of_rows)
 (TABLESPACE [=] tablespace_name)
 (NODEGROUP [=] node_group_id)

select_statement:
[IGNORE | REPLACE] [AS] SELECT ... (Some valid select statement)
```

```

CREATE TABLE employees_demo
( employee_id      NUMBER(6)
, first_name       VARCHAR2(20)
, last_name        VARCHAR2(25)  CONSTRAINT emp_last_name_nn_demo NOT NULL
, email            VARCHAR2(25)  CONSTRAINT emp_email_nn_demo      NOT NULL
, phone_number     VARCHAR2(20)
, hire_date        DATE DEFAULT SYSDATE CONSTRAINT emp_hire_date_nn_demo NOT NULL
, job_id           VARCHAR2(10)  CONSTRAINT emp_job_nn_demo        NOT NULL
, salary           NUMBER(8,2)   CONSTRAINT emp_salary_nn_demo     NOT NULL
, commission_pct   NUMBER(2,2)
, manager_id       NUMBER(6)
, department_id    NUMBER(4)
, dn               VARCHAR2(300)
, CONSTRAINT       emp_salary_min_demo CHECK (salary > 0)
, CONSTRAINT       emp_email_uk_demo   UNIQUE (email)
) ;

```

WERTEBEREICHE

ZEICHEN UND ZEICHENKETTEN

CHAR

**VARCHAR2 BZW
VARCHAR**

CHAR(10) 'abcdef '

VARCHAR2(10) 'abcdef '

DATUM
DATE
DD – MON – YY

ZEIT TIMESTAMP

ZEITZONEN

TIMESTAMP WITH TIME ZONE

TIMESTAMP WITH LOCAL TIME ZONE

NUMERISCHE WERTE

INTEGER
int

smallint

float

ORACLE NUMBER (precision, scale)

LOB

LARGE OBJECTS

CLOB

CHARACTER LARGE OBJECT

BLOB

BINARY LARGE OBJECT

CONSTRAINTS

PRIMÄRSCHLÜSSEL

```
CREATE TABLE Bücher (  
    ISBN char (13),  
    PRIMARY KEY (ISBN)  
)
```

```
CREATE TABLE Bücher (  
    ISBN char (13) PRIMARY KEY  
)
```

```
CREATE TABLE Bücher (  
    ISBN char (13),  
    CONSTRAINT PK_Bücher PRIMARY KEY (ISBN)  
)
```

UNIQUE

```
CREATE TABLE Bücher (  
    ID integer PRIMARY KEY,  
    ISBN char (13) UNIQUE  
)
```


FREMDSCHLÜSSEL

```
CREATE TABLE Bücher (  
    Verlagsname varchar (100),  
    FOREIGN KEY (Verlagsname) REFERENCES Verlage (Verlagsname)  
)  
CREATE TABLE Bücher (  
    Verlagsname varchar (100) REFERENCES Verlage (Verlagsname)  
)
```

REAKTIONEN AUF MODIFIKATIONEN

ON (<UPDATE> | <DELETE>)

<NO ACTION> | <SET NULL> | <SET DEFAULT> | <CASCADE>

```
CREATE TABLE supplier
(  supplier_id numeric(10) >not null,
  supplier_name varchar2(50) not null,
  contact_name varchar2(50),
  CONSTRAINT supplier_pk PRIMARY KEY (supplier_id)
);
```

```
CREATE TABLE products
(  product_id numeric(10) not null,
  supplier_id numeric(10) not null,
  CONSTRAINT fk_supplier
    FOREIGN KEY (supplier_id)
    REFERENCES supplier(supplier_id)
    ON DELETE CASCADE
);
```

INTEGRITÄTSBEDINGUNGEN

```
CREATE TABLE Bücher (  
    Jahr integer CHECK (Jahr BETWEEN 1980 AND 2050)  
)
```

DEFAULT-WERTE PRO SPALTE ANGEZEIGT WERDEN

```
CREATE TABLE Bücher (  
    Lieferbar SMALLINT(1) DEFAULT 1  
)
```

```
CREATE TABLE Bücher (  
    Bestelldatum DATE NOT NULL  
)
```

```
CREATE TABLE Bücher (  
    ISBN char (13),  
    PRIMARY KEY (ISBN),  
    Lieferbar smallint(1) default 1,  
    Jahr integer CHECK (Jahr BETWEEN 1980 AND 2050),  
    Verlagsname varchar (100),  
    PRIMARY KEY (ISBN),  
    FOREIGN KEY (Verlagsname) REFERENCES Verlage (Verlagsname)  
)
```

ALTER TABLE

DROP TABLE

SEQUENCE

```
CREATE SEQUENCE matrikel_seq;  
CREATE TABLE Student (  
    matrikelnummer INTEGER DEFAULT nextval('matrikel_seq'),  
    name VARCHAR(30)  
)
```

INDEX

```
CREATE INDEX schnelle_namen_suche ON Bücher (ISBN asc);
```



**DAS WAR'S FÜR
HEUTE**