

### Exercise 9: Snapshot Semantics

Give an example (table and INSERT statement) that demonstrates that INSERT statements have to be handled in two phases as well.

Solution:

```
INSERT INTO R  
  SELECT * FROM R;
```

If R contains a tuple, the select statement has to be executed independent of the insert statement otherwise the tuple is inserted recursively infinite times. The result of the select statement is inserted into a temporary table and then the temporary table is inserted into the target table.

### Exercise 10: Integrity constraints

Given the following Relational Schema

LESER (LSNR, NAME, VORNAME, WOHNORT, GEBDATUM)  
BUCH (ISBN, TITEL, AUTOR, SEITENZAHL, VERLEGT\_VON,  
 ERSCHINUNGSJAHR)  
VERLAG (VERLAGSNAME, VERLAGSORT)  
KATEGORIE (KATEGORIENAME, ENTHALTEN\_IN)  
EXEMPLAR (ISBN, EXEMPLARNUMMER, REGAL, POSITION)  
AUSLEIHE (LSNR, ISBN, EXEMPLARNUMMER, RÜCKGABEDATUM)  
BUCH\_KAT (ISBN, KATEGORIENAME)

a) Which problems appear if the schema is realized in SQL without consideration of the integrity constraints?

Solution:

- If primary keys are not used, there are data items which appear several times (e.g. several "Kategorien" with the same name)
- Dangling pointers after deleting or updating (e.g. books which do not exist anymore are still in the table "Exemplar")
- Semantically incorrect values in certain columns. Standard SQL allows no constraints in the data domain (e.g. birthdates in the future)

b) Which foreign keys are necessary and which strategies have to be applied when changing them? What are the reasons for the strategies you used?

Solution:

- E.g. Book: published\_by references publisher: ON DELETE SET NULL or NO ACTION, ON UPDATE NO ACTION oder CASCADE.

If a publisher is deleted the books which are published by this publisher should exist

without the publisher entry. One can also avoid the deletion. Renaming should be passed or prevented.

- E.g. Categories: included\_in references categorie: ON DELETE SET NULL, ON UPDATE NO ACTION or CASCADE

If a super-categorie is deleted, the sub-categorie should still exist without super-categorie. Renaming should be passed or prevented.

- E.g. Book\_Kat: ISBN references Book, categorie name references categorie. For both: ON DELETE CASCADE, ON UPDATE CASCADE. If a referenced tuple in a N:M relationship is deleted, all references should be deleted.
- E.g. Copy: ISBN references Book. For both: ON DELETE CASCADE, ON UPDATE CASCADE. If a tuple of the strong relation is deleted, the dependent tuples must be deleted as well.
- E.g. lending: LSNR references Reader, (ISBN, copy number) references copy. For both: ON DELETE CASCADE, ON UPDATE CASCADE. See Book\_Cat for the reasons.

- c) Which consequences have the deletion of a publisher and the changing of a reader number? Consider b)

Solution:

If no books of the publisher are available, the publisher is deleted. There are no other effects. If books are available, the deletion is either stopped (NO ACTION concerning the books) or the publisher of these books is set to NULL.

The modification of the reader number leads to a modification of the reader number at any lending of the reader. If the DBMS does not support ON UPDATE CASCADE, the renaming is prevented.

- d) Enforce that only readers who live in Zurich can be inserted!

In the definition (SQL DDL) of the table „reader“: (Wohnort in ('Zürich'))

- e) Enforce that a reader can only borrow 20 books! Give a solution using CHECK and a solution using Trigger.

With CHECK:

In AUSLEIHE:

```
CONSTRAINT CHECK (  
  NOT EXISTS (  
    SELECT COUNT(*) FROM Ausleihe  
    GROUP BY lsnr  
    HAVING COUNT(*) > 20)  
  )  
)
```

This is not supported in DB2. DB2 does not allow subqueries in check.

With TRIGGER:

```
CREATE TRIGGER AMAX
NO CASCADE
BEFORE INSERT
ON Ausleihe
REFERENCING NEW AS newr
FOR EACH ROW
MODE DB2SQL
WHEN ((Select count (*) from Ausleihe A where A.LSNR = newr.LSNR ) >= 20)
BEGIN ATOMIC
SIGNAL SQLSTATE '-1' SET Message_TEXT='Illegal Insert - too many books per
reader';
END
```

SQLSTATE is given as an example, a correct SQLSTATE should be defined as documented.

This Trigger treats only the case when the number of borrowed books is increased by INSERT („BEFORE INSERT“). It is still possible that another reader borrows the books and then the reader number is changed. In this case one needs another trigger with „on update“

This Trigger works in DB2.

### Exercise 11: Computation of the attribute hull

Given a relation (A, B, C, D, E, G) and the following eight functional dependencies  $F$

$AB \rightarrow C$	$C \rightarrow A$	$BC \rightarrow D$	$ACD \rightarrow B$
$D \rightarrow EG$	$BE \rightarrow C$	$CG \rightarrow BD$	$CE \rightarrow AG$

Provide the attribute hull  $a^+$  of  $(F, a)$

$a = \{BD\}$

Solution:

(„Erg“ is the result, o.Ä. = without change)

<u>Erg content</u>	<u>FD used</u>
Erg := BD	$D \rightarrow EG$
BDEG	$BE \rightarrow C$
BCDEG	$CG \rightarrow BD$
ohne Änderung	$CE \rightarrow AG$
ABCDEG	$AB \rightarrow C$

o.Ä.	C	→	A
o.Ä.	BC	→	D
o.Ä.	ACD	→	B
o.Ä.	D	→	EG
o.Ä.	BE	→	C
o.Ä.	CG	→	BD
o.Ä.	CE	→	AG

### Exercise 12:

Consider the following relation of orders:

Auftrag(  
 ProduktNr, ProduktName,  
 KundenNr, KundenName,  
 Datum,  
 Stückpreis,  
 Anzahl,  
 Nettopreis, Mehrwertsteuersatz, Bruttopreis)

Order (productNo, productName,  
 customerNo, customerName,  
 date,  
 unit price,  
 number of items,  
 net price, tax, gross price)

- Consider that the tax depends on the product (e.g. 8 % for books, 16% for luxury items)
- Several orders of the same customer are combined: if a customer orders several items a day, the order is combined- we only have one order per customer and per day

a) Determine all non-trivial functional dependencies in the relation “order”.

Solution:

-{ProduktNr} → {ProduktName, Stückpreis, Mehrwertsteuersatz}  
 -{KundenNr} → {KundenName}  
 -{ProduktNr, KundenNr, Datum} → {Anzahl}  
 -{Stückpreis, Anzahl} → {Nettopreis}  
 -{Nettopreis, Mehrwertsteuersatz} → {Bruttopreis}  
 {ProduktName} → {ProduktNr}  
 {KundenName} → {KundenNr}

b) Which are the candidates for keys?

Solution:

{ProduktNr, KundenNr, Datum}  
 {ProduktNr, KundenName, Datum}  
 {ProduktName, KundenName, Datum}  
 {ProduktName, KundenNr, Datum}

{ProduktNr, KundenNr, Datum}

=>

{ProduktNr, KundenNr, Datum}{ProduktName, Stückpreis, Mehrwertsteuersatz}

=>

{ProduktNr, KundenNr, Datum}{ProduktName, Stückpreis, Mehrwertsteuersatz}  
{KundenName}

=>

{ProduktNr, KundenNr, Datum}{ProduktName, Stückpreis, Mehrwertsteuersatz}  
{ KundenName}{Anzahl}

=>

{ProduktNr, KundenNr, Datum}{ProduktName, Stückpreis, Mehrwertsteuersatz}  
{ KundenName}{Anzahl}{Nettopreis}

=>

{ProduktNr, KundenNr, Datum}{ProduktName, Stückpreis, Mehrwertsteuersatz}  
{ KundenName}{Anzahl}{Nettopreis}{Bruttopreis}