

12 Anhang

Änderung- und Versionsverfolgung		
Ursprung: anhang06_7vers1.rtf 06.07.1999 17:48		
Datum	Action	Bemerkung
06.07.1999	Erweitern	Kap. 12.4 Sonstiges um Ablaufplan/diagramm zur Testfallermittlung *vers2
07.07.1999	Ändern	Codd (Hr) *vers1
08.07.1999	Erweitern	Begriffe Seite 8: PK; FK; ref. Int.; Normalisierung etc *vers1
09.07.1999	Ändern	T_AKTIV eingefuegt *vers1
19.07.1999	Backup	*19_7vers1
19.07.1999	Ändern	*19_7vers2 Tabellen und Bilder ändern
19.07.1999	Erweitern	*19_7vers3 2 Tabellen ins DB-Schema
22.07.1999	Ändern	*22_7VERS1 RS UND DIVERSE AENDERUNGEN

12.1 Funktion und Arbeitsweise einer relationalen Datenbank

Typische Anwendungen für relationale Datenbanken sind unter anderem die allgemeine Datenhaltung (Adressen, Bücher, Videos, CDs usw.), Lagerverwaltung, Bestellwesen, Fakturierung sowie Wissensspeicher, Lexika etc. In einem Datenbanksystem können mehrere unterschiedliche unabhängige Datenbanken betrieben werden. Für jede dieser Datenbanken ist ein eigener Satz von Prozessen zuständig (Instanzen).

[relDB]

12.1.1 Anforderungen an ein Datenbanksystem

Im folgenden sind die grundlegenden Anforderungen an ein relationales Datenbankmanagementsystem (RDBMS) aufgeführt:

- physische Datenunabhängigkeit zwischen Programmen und Daten,
- unabhängig von Hardware (Hardwareherstellern),
- einfache Handhabung,
- sicher und zuverlässig,
- normierte Schnittstelle(n),
- offene Standards,
- Mehrbenutzerbetrieb,
- kleine Reaktionszeiten,

- Unterstützung von nationalen Sprachen/ Zeichen.

Oracle erfüllt diese Anforderungen mit einem hohen Grad an Portabilität. Für nahezu jede Plattform wird das RDBMS angeboten. Die Applikationen können aufgrund der Trennung von GUI und Programm leicht portiert werden.

12.1.2 Funktionalität relationaler Datenbank-Systeme

Die folgenden 12 Regeln gehen auf E.F. Codd zurück und beschreiben einzelne Funktionalitäten, die von relationalen Datenbanksystemen erfüllt werden sollten. Die Regeln gestatten die Bewertung der "Relationalität" gegebener Datenbank-Implementierungen.

Informationsregel	In einer relationalen Datenbank werden alle Informationen ausschließlich logisch, durch Werte in Tabellen repräsentiert. Tabellen ohne einen einzigen, oder nur mit einem Eintrag, sind gültige Tabellen.
Zugriffsregel	Auf jeden einzelnen, atomaren Datenwert einer relationalen Datenbank kann durch entsprechende Angabe des Tabellennamens, des spaltenadressierenden Spaltennamens und des zeilenadressierenden Primärschlüssel jederzeit logisch zugegriffen werden.
NULL-Wertbehandlung	Unabhängig vom Datentyp werden fehlende oder nicht verfügbare atomare Datenwerte nach außen durch einen Leerwert-Indikator repräsentiert. Für die Manipulation und das Retrieval fehlender Datenwerte stehen unterstützende Funktionen bereit.
Relationale, dynamische Metadaten	Datenbankobjekt-Beschreibungs- und –Verwaltungs-Informationen werden wie Benutzerdaten auf logischer Ebene geführt und stehen zugriffberechtigten Anwendern über die Standard Datenbank-Sprachmittel zur Verfügung. Der Datenbank-Katalog (Data Dictionary) muß als dynamischer, relation-orientierter Online-Katalog implementiert sein.
Datenbanksprachen	Es existiert zumindest eine syntaktisch wohl- definierte Sprache, die Teilsprachen zur Datendefinition, zur Viewdefinition, zur interaktiven und programmgesteuerten Datenmanipulation, zur Benutzerauthorisierung, zur Transaktionssteuerung und zur Definition von Integritätsbedingungen und –regeln umfaßt. Danaben sind weitere Sprachen zugelassen.
Datenänderungen im "View"	Daten, die theoretisch in einer View geändert werden könnten, sind tatsächlich manipulierbar.
Mengenorientierte Manipulationen	INSERT-, UPDATE- und DELETE-Befehle operieren auf Tabellen mengenorientiert.
Physische Datenunabhängigkeit	Physische Datenspeicherung und tatsächliche Zugriffe sind Anwendungsprogrammen und Benutzerzugriffen verborgen. Anwendungsprogramme und Dialoganwendungen werden von einer Änderung der physischen Speicherung oder einem Wechsel der Zugriffsmethoden logisch nicht tangiert.
Logische Datenunabhängigkeit	Informationsbewahrende Änderungen implementierter Tabellenstrukturen werden gegenüber

	Anwendungsprogrammen und Dialoganwendungen transparent gehalten.
Integritätsunabhängigkeit	Integritätsbedingungen und –regeln werden in der Datenbanksprache verfaßt und im Datenbank-Katalog hinterlegt. Die Einhaltung der Regeln stellt das relationale Datenbanksystem sicher. Integritätsregeln sollten nicht auf Anwendungsebene implementiert werden müssen.
Verteilungsunabhängigkeit	Der Verteilung von Tabellen auf verschiedene Datenbanken, eine eventuell nötige Transaktionssplittung und verteilte Verarbeitung muß gegenüber Anwendungsprogrammen und Benutzerzugriffen transparent sein.
Non- Subversions-Regel	Der Zugriff auf Datenbankinhalte darf nur über die dedizierte Datenbanksprache erfolgen. Definitionen, Sicherungsbeschreibungen und Integritätsregeln dürfen nicht durch andere Sprachmittel, die eine Schnittstelle zum Datenbanksystem besitzen, umgangen oder verletzt werden.

Tabelle 12.1-1: Regeln zur Bewertung der Relationalität

Zu den Komponenten eines RDBMS gehören die Netzwerkanbindung, die integrierten Entwicklungswerkzeuge, Applikationen, Backup/Recovery-Funktionen, Import/Export-Möglichkeiten sowie das Administrationsinterface.

12.1.3 Datenbankobjekte

Es gibt verschiedene Datenbankobjekte, die ein Datenbanksystem zur Verfügung stellt:

Relationales Modell	Tabellendarstellung	Datenverarbeitung
Relation	Tabelle	Recordset
Attribut	Spalte	Feld
Tupel	Zeile	Datensatz, Record
Attributname	Spaltenüberschrift	Feldname
Grad der Relation	Spaltenzahl	Feldanzahl
Kardinalität	Zeilenzahl	Zahl der Datensätze

Tabelle 12.1-1: Begriffe des relationalen Modells [DBLin]

Objekt (Englisch)	Funktion
Table	Eine Tabelle repräsentiert einen Entitätstyp und faßt dessen Attribute zusammen. Sie dient zur Aufnahme von Entitäten (den Daten).
Row	Eine Row beinhaltet einen Datensatz (Zeile) einer Tabelle.
Columns	Eine Spalte repräsentiert ein Attribut. In jeder Row besitzt eine Spalte einen (Feld-) Wert.
View	Sichten auf Tabellen entweder zur Zusammenfassung von ein oder mehreren Tabellen (JOIN) oder zum Verbergen von Informationen
Sequence	Nummerngenerator zur Erzeugung von (eindeutigen) Schlüsseln
Index	Ein Index ist eine Zugriffsstruktur für den schnellen Zugriff über die Werte einer oder mehrerer Spalten auf die Rows einer Tabelle. Bei ORACLE ist ein Index als B*-Baum realisiert.
Synonym	Aliasname für DB-Objekt, vergleichbar mit einem "Link" im Filesystem.
Grants	User-Privilegien
Roles	Rolle; Gruppe von Usern

Die Hauptmerkmale von SQL, als Vertreter für eine Sprache der 4. Generation, sind:

- Einheitlich für alle Benutzergruppen,
- nicht prozedural,
- Sprache der 4. Generation (4GL),
- Relational (ein Feld oder ganze Tabelle),
- Mengenorientiert.

Klassifizierung von Datenbanksprachen

nach der Funktion:

DDL	Data Definition Language Datenbeschreibungssprache
DML	Data Manipulation Language Datenmanipulationssprache
SSL	Storage Structure Language Speicherstruktursprache
DCL	Data Control Language Datenkontrollsprache

nach der Art der Benutzung:

selbstständig	-	selfcontained
nichtselbstständig	-	depended

nach dem Prinzip der Formulierung der Anforderung:

prozedural	-	wie soll das Ergebnis erreicht werden
deskriptiv	-	was für ein Ergebnis soll erreicht werden

12.1.4 Datenintegrität

Entitäts- Integrität	Primärschlüssel dürfen keine Null-Werte enthalten und müssen als Identifikation eindeutig sein.
-------------------------	---

Referentielle Integrität	Fremdschlüssel einer Tabelle sind entweder Null oder sie stimmen mit einem Primärschlüssel, der über den zugehörigen Referenz-Pfad angesprochenen Tabellen, überein.
-----------------------------	--

Domänen- Integrität	Spalten (einer Tabelle) dürfen nur Werte annehmen, die den ihnen zugeordneten Domänen entstammen.
------------------------	---

Referenzielle Integrität:

Das DBMS überwacht die vom Nutzer (DB-Designer, DBA) definierten Abhängigkeiten zwischen den Datenwerten verschiedener Tabellen. Voraussetzung dafür ist, daß diese Abhängigkeiten

1. mit der DDL beschrieben werden können,

d.h.	Primärschlüssel	-	PRIMARY KEY
	Fremdschlüssel	-	FOREIGN KEY

definiert werden können und

2. festgelegt werden kann, was bei Löschung eines zugeordneten Primärschlüssels geschehen soll.

Die Löschung eines Primärschlüsselwertes kann:

1. verweigert werden, so lange noch ein zugehöriger Fremdschlüssel existiert (RESTRICT),
2. weitergereicht werden (CASCADES), d.h. die Datensätze mit den zugehörigen Fremdschlüsseln werden ebenfalls gelöscht,
3. dazu führen, daß der Fremdschlüssel den Wert Null zugewiesen bekommt (SET NULL).

Datenintegrität:

Die Datenbank muß in Ordnung (integer) sein. Unterscheidung von drei Komplexen

▪ Datenkonsistenz

- logischer Aspekt der Integrität
- Widerspruchsfreiheit, log. Richtigkeit
- Regeln (Konsistenzbedingungen) beim Entwurf der Datenbank festlegen
- Transaktionskonzept

▪ Datensicherheit

- physischer Aspekt der Integrität
- technische und organisatorische Maßnahmen, um die Daten vor physischer Zerstörung, Verlust und Beschädigung zu schützen
- Backup und Recovery- Verfahren

▪ Datenschutz

- ethnischer Aspekt der Integrität
- Verhinderung der mißbräuchlichen Verwendung der Daten
- Rechtesystem (GRANT und REVOKE)

Die Sicherung der Einheitlichkeit der Daten, die Datenkonsistenz, wird durch die folgenden Mechanismen implementiert:

- Constraints (Eingaben in Spalten erzwingen bzw. begrenzten Plausibilitätsprüfungen unterziehen)
 - NOT NULL
 - DEFAULT
 - PRIMARY KEY
 - UNIQUE
 - FOREIGN KEY / REFERENCES
 - CHECK
- Transaktionskonzept
- Locking
 - Multi-User-Betrieb
 - Zugriffssperren bei Änderungen
 - Transaktionen (2 Checkpoints; Start und End of Transaction)
 - Deadlock

Locking (Zugriffssperren)

- Zugriffssperre auf Daten, die während einer Transaktion berührt werden
- das System der Zugriffssperren ist für den normalen Anwender unsichtbar

- Explizite Zugriffssperren sind möglich
- Aufheben der Zugriffssperren durch COMMIT oder ROLLBACK
- SET TRANSACTION READ ONLY
- Row Level Locking – Zugriffssperren auf Zeilenebene
- Table Level Locking – Zugriffssperren auf Tabellenebene

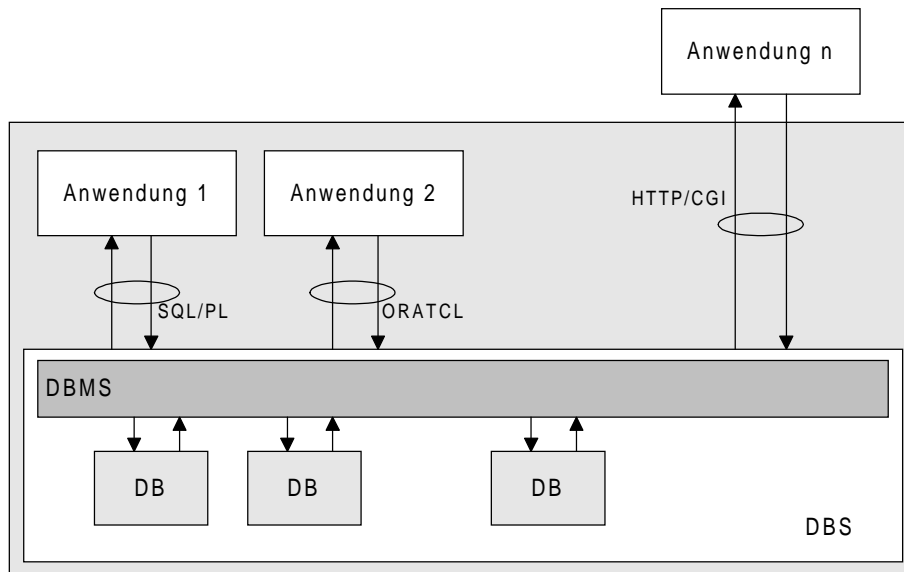


Abbildung 12.1-1: Datenverwaltung mit Datenbanken [DBLin]

Um in einem Datenbankmodell eine möglichst geringe Redundanz (Wiederholung von abhängigen Daten) zu sichern, werden die Relationen normalisiert. Ziel der Normalisierung ist, die Attribute so zu Entitätsmengen bzw. Relationen zu ordnen, daß innerhalb einer Relation keine Redundanzen auftreten.

Mit verschiedenen Schlüsseln (key) werden die Datensätze einer Datenbank organisiert. Ein Identifikationsschlüssel (primary key; PK; Primärschlüssel) dient dazu, einen Datensatz eindeutig zu identifizieren.

Um in einer Relation ein Attribut zu bezeichnen, dessen Wert vom aktuellen Inhalt dieses Attributs in einer anderen Relation abhängt, wird der Begriff des Fremdschlüssels (foreign key; FK) verwendet. Ein Fremdschlüssel in einer Relation ist ein Attribut (oder eine Attributkombination), welches dem Identifikationsschlüssel in einer anderen Relation entspricht. Damit läßt sich eine hierarchische Beziehung zwischen Relationen definieren.

Eine relationale Datenbank darf nur referenzierende Fremdschlüssel enthalten. Das heißt, nur solche FK, denen, wenn sie nicht NULL sind, ein Wert /eine Kombination von Werten im Primärschlüssel der referenzierenden Zielrelation entspricht.

[DBLin]

12.1.4.1 Beziehungen zwischen den Objektklassen

Symbol	Beziehung	Assoziationstyp	Beispiel
1	zu-eins	Einfache Assoziation	Jeder Testfall gehört zu einem Testobjekt.
c	zu-eins ODER keins	Konditionelle Assoziation	Jeder Testfall ist einem oder keinem Tester zugeordnet.
m	zu-mindestens- eins	Multiple Assoziation	Jedes Projekt hat mindestens einen Auftraggeber.
mc	zu-keins, eins ODER mehreren	Multiple- konditionelle Assoziation	Jeder Tester ist keinem, einem oder mehreren Testfällen zugeordnet.

Tabelle 12.1-1: Assoziationstypen in ER-Modellen [DBLin]

Entity Relationship Diagramm Konventionen

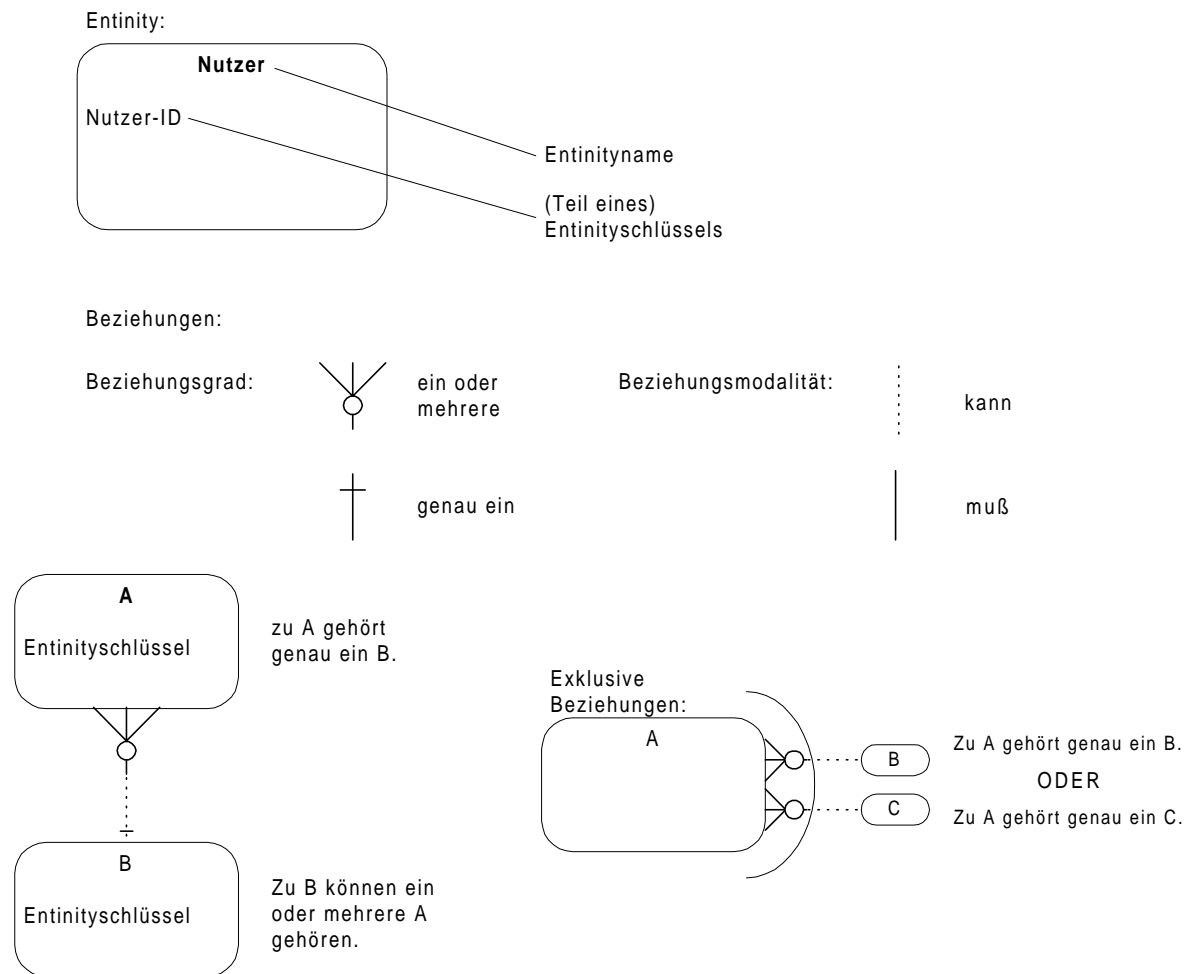


Abbildung 12.1-1: Entity Relationship Diagramm Konventionen

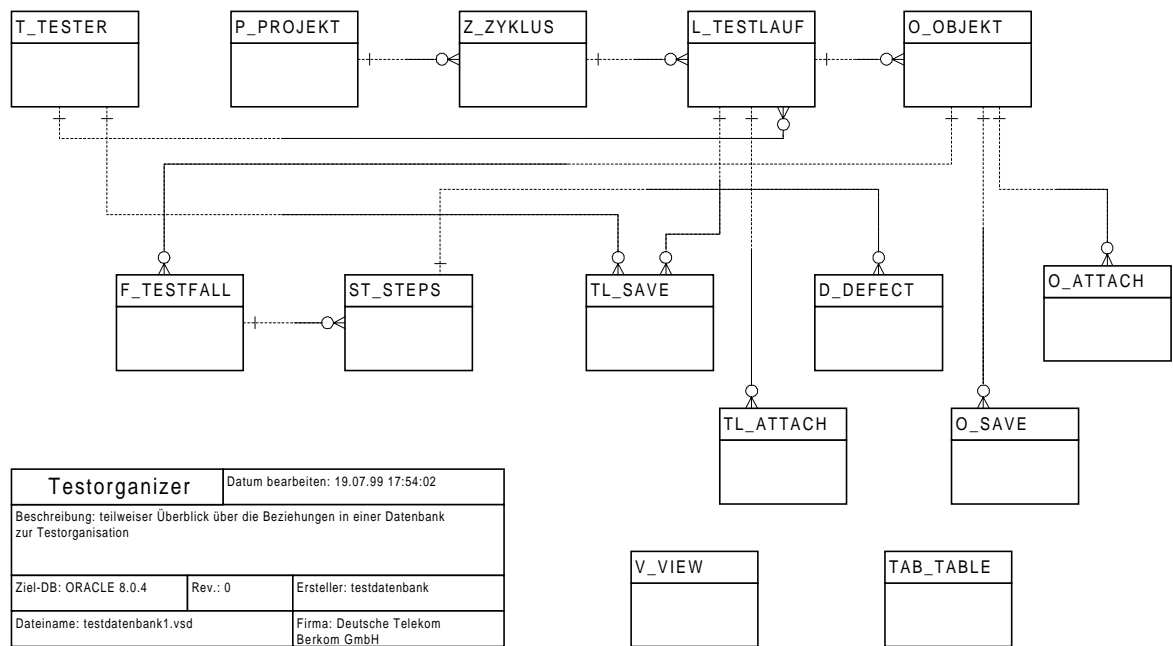


Abbildung 12.1-2: Relationship in der Testdatenbank

Die Abkürzungen in der Spalte der Constraints bedeuten:

p = Primary Key

f = Foreign Key

u = unique

c = check

n = not Null

d = default

o = on delete cascade

Name:	Tester Table				
SHORT NAME:	T_TESTER				
COMMENTS:	Die Grundinformationen über die Mitarbeiter im ITB.				
FIELD	Type	Length	Constraints	Description	Programoptions
T_ID	Number	3	p,n	(PK)	Not null
T_NNAME	Varchar2	40	n	Nachname des Testers	Not null
T_AKTIV	Varchar2	1	n	A=Aktiv; B=Inaktiv	Standard A
T_VNAME	Varchar2	30	n	Vorname des Testers	Not null
T_EMAIL	Varchar2	60		Die eMail des Testers	Optional
T_PASSWORT	Varchar2	9	n	Das Klartextpasswort	Not null
T_ABTEILUNG	Varchar2	30		Abteilung des Testers	Optional

Tabelle 12.1-2: Tester_Table

Name:	Projekt Table				
Short Name:	P_PROJEKT				
Comments:	Die Grundinformationen über das entsprechende Projekt.				
Field	Type	Length	Constraints	Description	Programoptions
P_ID	Number	3	p,n	(PK)	Not null
P_PROJEKTNAME	Varchar2	40	n	Projektname	Not null
P_AKTIV	Varchar2	1	n	A=Aktiv; B=Inaktiv	Standard A
P_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
P_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
P_PROJEKTGEBER	Varchar2	40	n	Auftraggeber	
P_BEGIN	Date		n	Beginn der Tests am Projekt	Not null
P_END	Date			Ende der Tests am Projekt	

Tabelle 12.1-3: Projekt_Table

Name:	Zyklus Table				
Short Name:	Z_ZYKLUS				
Comments:	Die unterschiedlichen Testzyklen, in denen getestet wird.				
Field	Type	Length	Constraints	Description	Programoptions
Z_ID	Number	6	p,n	PK	Not null
Z_PID	Number	3	f,n	(FK—P_ID) Projekt	Not null
Z_AKTIV	Varchar2	1	n	Aktivierung und Inaktivierung des Zyklus möglich.	Standard A
Z_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
Z_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
Z_ZYKLUS	Varchar2	20	n	Zyklusname	Not null
Z_DESCR	Varchar2	256		Eine Beschreibung des Zyklus.	

Tabelle 12.1-4: Zyklus_Table

Name:	Testlauf Table				
Short Name:	L_TESTLAUF				
Comments:	Die den Testzyklen zugehörigen Testläufe.				
Field	Type	Length	Constraints	Description	Programoptions
L_ID	Number	6	p,n	PK	Not null
L_LAUF	Varchar2	20	n	Der Testlaufname.	Not null
L_AKTIV	Varchar2	1	n	Aktivierung und Inaktivierung des Laufes.	Standard A
L_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
L_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
L_BEARBEITER	Number	3	n	(FK—T_ID) Bearbeiter des TL.	Not null
L_DESCR	Varchar2	256		Eine TL-Beschreibung.	

Tabelle 12.1-5: Testlauf_Table

Name:	Objekt Table				
Short Name:	O_OBJEKT				
Comments:	Die Informationen über die Testobjekte in den zugehörigen Testläufen.				
Field	Type	Length	Constraints	Description	Programoptions
O_ID	Number	15	p,n	Testobjektnummer (PK)	Not null
O_TO	Varchar2	40	n	Name des Testobjekts	Not null
O_AKTIV	Varchar2	1	n	Aktivierung oder Inaktivierung des Objekts.	Standard A
O_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
O_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
O_PROJEKT	Number	3	f,n	(FK—P_ID) Das übergeordnete Projekt des TO.	Not null
O_BEARBEITER	Number	3	f,n	(FK—T_ID) Bearbeiter des TO.	Not null
O_BEGINN	Date		n	Beginn der Testobjektabgrenzung	
O_SOLLENDE	Date			Voraussichtliches Ende der Tests an dem TO.	
O_ENDE	Date			Abschluß der Tests an dem TO.	
O_STATUS	Varchar2	1	n	Derzeitiger Testzustand in welchem sich das TO befindet.	A=Testdatendefinition B=Test; C=ReTest; D=ReDefinition; F=Test; G=Abschluß
O_KOMMENTAR	Varchar2	512		Optionaler Kommentar zu dem TO.	

Name:	Objekt Table				
Short Name:	O_OBJEKT				
Comments:	Die Informationen über die Testobjekte in den zugehörigen Testläufen.				
Field	Type	Length	Constraints	Description	Programoptions
O_BESCHR	Varchar2	512	n	Kurzbeschreibung des TO	
O_ZYKLUS	Varchar2	20	n	Testzyklus-Versionsnummer	Not null

Tabelle 12.1-6: Objekt_Table

Name:	Attachment Table				
Short Name:	TL_ATTACH				
Comments:	Die den Testläufen zugeordneten Attachments.				
Field	Type	Length	Constraints	Description	Programoptions
TLA_ID	Number	5	p,n	(pk) Der Primär-Key der Attachments.	
TLA_DESCR	Varchar2	512		Beschreibung des Attachments. A=C-Scripte B=LR-Runner C=WR-Scripte; D=TCL-Scripte;	Optional
TLA_AKTIV	Varchar2	1	n	Aktivierung und Inaktivierung des Attachments	Standard A
TLA_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
TLA_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
TLA_FILE	Varchar2	256	n	Der lokale Speicherort im Filesystem der DB.	
TLA_FROM	Number	15	f,n	(FK→ L_ID) Der zum File gehörende Testlauf.	F=BAT; G=binär; H=ASCII;
TLA_ATTACHTYP	Varchar2	1	n	Der Typ des Attachments. Auswahl	

Tabelle 12.1-7: TLAttachment_Table

Name:	Testfall Table				
Short Name:	F_TESTFALL				
Comments:	Informationen zu den Testfällen innerhalb der entsprechenden TO/Projekt-Kombination.				
Field	Type	Length	Constraints	Description	Programoptions
F_ID	Number	15	p,n	Der Primärkey der Testfalltables.	Automatisch
F_TESTFALL	Varchar2	50	n	Die char-Beschreibung des Testfalls.	Not null
F_AKTIV	Varchar2	1	n	Testfall aktiv?	Standard A
F_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
F_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
F_TESTOBJEKT	Varchar2	40	f,n	(FK—O_ID) Das übergeordnete Testobjekt.	Not null
F_KRITIKALITAET	Char	1	n	Der Charakter bestimmt die Kritikalität des Testfalls.	Not null A = Hoch B = Mittel C = Niedrig
F_BEARBEITER	Varchar2	40	f,n	(FK—T_ID) Die ID des Testers der an diesem Testfall arbeitet.	Not null
F_SOLL	Varchar2	512	n	Das erwartete Ergebnis des Testlaufs.	Not null
F_IST	Varchar2	512		Das reale Testergebnis.	

Tabelle 12.1-8: Testfall_Table

Name:	Save Table				
Short Name:	TL_SAVE				
Comments:	Die Sicherungsinformationen des Archivierungssystems bzgl. des TL.				
Field	Type	Length	Constraints	Description	Programoptions
TLS_ID	Number	17	p,n	(PK)	Not null
TLS_SAVELAUF	Varchar2	40	n	Name des Sicherungslaufs entsprechend der Namenskonvention	(halb-)automatisch ?Namenskonvention?
TLS_AKTIV	Varchar2	1	n	Sicherungslauf aktiv?	Standard A
TLS_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
TLS_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
TLS_SAVETESTFALL	Number	15	f,n	(FK—L_ID) Bezeichnung des TF der gesichert wird.	Not null / Automatisch
TLS_TYP	Varchar2	1	n	Die Art der Sicherung	A=PD B=SD
TLS_SAVEINITIATOR	Number	3	f,n	(FK—T_ID) Name des Ausführenden.	Not null / Automatisch
TLS_SAVEDATUM	Date		n	Datum der Sicherung	Not null
TLS_SAVEKOMMENTAR	Varchar2	512		Kommentar zur Sicherung. evtl. mehrere Sicherungen am selben Tag.	evtl. mehrere Sicherungen am selben Tag

Tabelle 12.1-9: TL_Save_Table

Name:	Steps Table				
Short Name:	ST_STEPS				
Comments:	Informationen zu den einzelnen Steps innerhalb eines Testfalls.				
Field	Type	Length	Constraints	Description	Programoptions
ST_ID	Number	10	p,n	Der Primär-Key der Steps Tabelle.	Not null
ST_TFSTEP	Number	15	f,n	(FK—F_ID) Der Testfall in welchem der Teststep ausgeführt wird.	Not null
ST_AKTIV	Varchar2	1	n	Step aktiv?	Standard A
ST_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
ST_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
ST_NUMBER	Number	3	n	Die Nummer des Steps innerhalb des Testfalls.	Not null
ST_ACTION	Varchar2	40	n	Die auszuführende Aktion in diesem Step. (Kurzbeschreibung)	Not null
ST_DESCRIBE	Varchar2	512		Eine Beschreibung des auszuführenden Steps.	

Tabelle 12.1-10: Steps_Table

Name:	Defect Table				
Short Name:	D_DEFECT				
Comments:	Die den Testfällen zugeordneten Defects.				
Field	Type	Length	Constraints	Description	Programoptions
D_ID	Number	6	p,n	PK	Not null
D_STEPID	Number	10	f	(FK—ST_ID) Der fehlerhafte Step.	Not null
D_AKTIV	Varchar2	1	n	Defect aktiv?	Standard A
D_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
D_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
D_SUBJEKT	Varchar2	50	n	Name des Defects.	Not null
D_DESCR	Varchar2	512		Beschreibung des Defects.	
D_REPRO	Varchar2	1	n	Reproduzierbar ?	Not null
D_PRIO	Varchar2	1	n	Priorität A=Formfehler; B=Didaktik; C=Systemrelevant	Not null
D_VOM	Date		n	Datum an welchem der Defect gefunden wurde.	Not null
D_VON	Number	6	f	(FK—T_ID)	Not null
D_STATUS	Varchar2	1	n	Der Status des Defects A=neu; B=gemeldet; C=behoben.	Not null

Tabelle 12.1-11: Defects_Table

Name:	Attachment Table				
Short Name:	O_ATTACH				
Comments:	Die den Objekten zugeordneten Attachments.				
Field	Type	Length	Constraints	Description	Programoptions
OA_ID	Number	5	p,n	(pk) Der Primär-Key der Attachments.	
OA_DESCR	Varchar2	512		Beschreibung des Attachments. A=C-Scripte B=LR-Runner C=WR-Scripte; D=TCL-Scripte;	Optional
OA_AKTIV	Varchar2	1	n	Aktivierung und Inaktivierung des Attachments	Standard A
OA_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
OA_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
OA_FILE	Varchar2	256	n	Der lokale Speicherort im Filesystem der DB.	
OA_FROM	Number	15	f,n	(FK→ O_ID) Der zum File gehörende Testlauf.	F=BAT; G=binär; H=ASCII;
OA_ATTACHTYP	Varchar2	1	n	Der Typ des Attachments. Auswahl	

Tabelle 12.1-12: O_Attach_Table

Name:	Save Table				
Short Name:	O_SAVE				
Comments:	Die Sicherungsinformationen des Archivierungssystems bzgl. der TO.				
Field	Type	Length	Constraints	Description	Programoptions
OS_ID	Number	17	p,n	(PK)	Not null
OS_SAVELAUF	Varchar2	40	n	Name des Sicherungslaufs entsprechend der Namenskonvention	(halb-)automatisch ?Namenskonvention?
OS_AKTIV	Varchar2	1	n	Sicherungslauf aktiv?	Standard A
OS_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
OS_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
OS_SAVETESTFALL	Number	15	f,n	(FK—O_ID) Bezeichnung des TF der gesichert wird.	Not null / Automatisch
OS_TYP	Varchar2	1	n	Die Art der Sicherung	A=PD B=SD
OS_SAVEINITIATOR	Number	3	f,n	(FK—T_ID) Name des Ausführenden.	Not null / Automatisch
OS_SAVEDATUM	Date		n	Datum der Sicherung	Not null
OS_SAVEKOMMENTAR	Varchar2	512		Kommentar zur Sicherung. evtl. mehrere Sicherungen am selben Tag.	evtl. mehrere Sicherungen am selben Tag

Tabelle 12.1-13: O_Save_Table

Name:	View Table				
Short Name:	V_VIEW				
Comments:	Die den Testfällen zugeordneten Defects.				
Field	Type	Length	Constraints	Description	Programoptions
V_ID	Number	6	p,n	PK	Not null
V_KOMMAND	Varchar2	256	n	Der View als ASCII	
V_AKTIV	Varchar2	1	n	View aktiv?	Standard A
V_WER	Number	3	f,n	Wer hat in dieser Zeile zuletzt geschrieben?	Not null
V_WANN	Date		n	Wann wurde in dieser Zeile zuletzt geschrieben?	Not null
V_DESCR	Varchar2	256		Eine View-Beschreibung.	

Tabelle 12.1-14: View_Table

NAME:	Table Table				
SHORT NAME:	TAB_TABLE				
COMMENTS:	Die notwendigen Privilegien zur Verwaltung der Testumgebung über die Tables.				
FIELD	Type	Length	Constraints	Description	Programoptions
TAB_ID	Number	2	p,n	PK	Not null
TAB_NAME	Varchar2	20	n	Der Name des Tables innerhalb Oracles.	Not null
TAB_PRIVILEG	Number	1	n	Die Privilegienklasse des Testers für die Action.	Not null

Tabelle 12.1-15: Table_Table

12.2 Implementation der Backuplösung

12.2.1 Umgebung

Server der jeweiligen Projekte:

qitb29	HTML-Server, P-PRO 200; NT 4.0; 6GByte HDD
qitb28	LDAP-Server, P-PRO 200; NT 4.0; 6GByte HDD

Clients der jeweiligen Projekte:

qitb24	Test-Client, Pentium 233; WIN 95; 3GByte HDD
qitb25	Test-Client, Pentium 233; WIN 95; 3GByte HDD
qitb26	Test-Client, Pentium 233; WIN 95; 3GByte HDD

Server der Testumgebung:

qitb99	HTML-Server, P-PRO 200; NT 4.0; 6GByte HDD
qitb12	FTP;PDC;Oracle-Server; PII 400; NT 4.0; 16GByte HDD

Clients der Testumgebung:

qitb14	Test-Client, Pentium 233; WIN 95; 3GByte HDD
qitb15	Test-Client, Pentium 233; WIN 95; 3GByte HDD
qitb16	Test-Client, Pentium 233; WIN 95; 3GByte HDD

Backup / Restore- Server

qitb04	Backup-Server, SUN Sparc-Station 20, Solaris 2.4, 64 MB Speicher, 125 MHz Ross HyperSparc, S-Bus Ethernet-Adapter mit 10/100 MBit/s.
--------	--

12.2.2 Installation der Backup-Software:

Die Sicherung erfolgt auf Basis von:

- NetWorker LeGaTo Archivierungssystem
- LibraryXpress, LXB2X10 von Overland Data; DLT7000 Wechsler

Folgende Installationen wurden vorgenommen:

Node	SBU	Lage /nsr	DMO
qitb01...99 (Clients)	Client	c:/Programme/Networker	
qitb13 (Fileserver;Oracle-DB)	Client	c:/Programme/Networker	X
qitb04 (Arch.-Server)	Server; Storage Node; Client; Man Pages	/opt/nsr	

DMO → Database Module for Oracle (SDMO) 2.1

12.2.3 Randbedingungen

Hinsichtlich der Datensicherung besteht das Intranet-Testbed aus 2 verschiedenen voneinander unabhängigen Sicherungsaufgaben. Für alle Rechner des ITB sollten für den Fall eines Crashes zu später beschriebenen Zeitpunkten Backups durchgeführt werden. Weiterhin werden von den an der aktiven Testung beteiligten Rechnern je nach Testobjekt und schreibendem Zugriff auf den Primär- und Sekundärdatenbestand incrementelle und vollständige Archivierungsläufe durchgeführt. Diese Archivierungsläufe haben keine festgelegte Livetime. Die Löschung der Archivierungsläufe erfolgt manuell durch den Testleiter.

Das Zeitfenster für die Backupläufe liegt zwischen 21.00-6.00 Uhr.

Für die Archivierungsläufe ist kein Zeitfenster gegeben, da sie manuell je nach Bedarf durch den Tester gestartet werden. Ebenso ist die ständig verfügbare Notwendigkeit eines Restore's in dem Zeitfenster zwischen 6.00-21.00 Uhr zu beachten. Full-Backups sollten grundsätzlich an Wochenenden laufen. Gravierend sind Ausfälle einzelner Rechner aufgrund der dezentralen Softwarehaltung nicht.

Die Testergebnisse in der Oracle-DB sollten täglich gesichert werden. Dies kann wahlweise durch die Verwendung eines "Networker Database Module for Oracle" oder durch Stoppen des Datenbank-Management-Systems (DBMS) und anschließende Sicherung auf Filesystemebene erfolgen. Die Sicherung des File-Servers sollte monatlich in einem 'full-backup' und die Datenverzeichnisse wöchentlich in einem 'incrementell-backup' erfolgen. Die Aufbewahrungsfristen (retention policy) der Backupdaten ist nicht spezifiziert. Ein Zugriff auf die gesicherten Daten erfolgt nur im Störfall. Die Cartridges aus dem Wechsler

werden regelmäßig ausgetauscht und archiviert. Die Einträge im 'Media Index' werden nach Ablauf der 'retention policy' (? Monate) entfernt. Die 'retention policy' entspricht damit nicht der tatsächlichen Aufbewahrungsfrist, sondern gewährleistet lediglich die Entfernung der Einträge aus dem 'Media Index'. Durch einen Bootstrap-Ausdruck, einscannen und labeln der Bänder ist auch ein Restore von Dateien möglich, deren Eintrag im 'Media Index' gelöscht wurden. Somit wird die regelmäßige Auslagerung der Cartridges zur wesentlichen Voraussetzung für die ordnungsgemäße Einhaltung der Aufbewahrungsfristen.

12.2.4 Scheduling

Die Voll-Sicherung (full-backups) der Projekt-Clients erfolgt monatlich.

Die Voll-Sicherung (full-backups) der Projekt-Server erfolgt wöchentlich

Die Voll-Sicherung (full-backups) der Testumgebungs-Clients erfolgt monatlich.

Die Voll-Sicherung (full-backups) der Testumgebungs-Server erfolgt wöchentlich.

Die incrementellen backups der Projekt-Server erfolgen täglich.

Die incrementellen backups der Testumgebungs-Server erfolgen täglich.

Die Sicherungen werden sequentiell über qitb04:/sbin/backup.sh aufgerufen. Der Aufruf des Scripts erfolgt via crontab:qitb04.

Die Archivierung wird getrennt geplant und manuell durchgeführt.

12.2.5 Zeitlicher Ablauf

Das Script /sbin/backup.sh dient zur Darstellung des zeitlichen Rahmens.

```
/usr/bin/nsr/savegrp [-c CLIENT_a] GRUPPE_1
```

```
/usr/bin/nsr/savegrp [-c CLIENT_b] GRUPPE_2
```

```
#oracle offline-sicherung
```

```
#shutdown oracle
```

```
/usr/bin/nsr/savegrp [-c CLIENT_c] GRUPPE_3
```

```
#startup oracle
```

```
#oracle online-sicherung
```

```
<rman>
```

#druck bootstrap

#check client und mediaindex

#auswertung ergebnis und meldung an operator (eMail)

12.3 Sonstiges

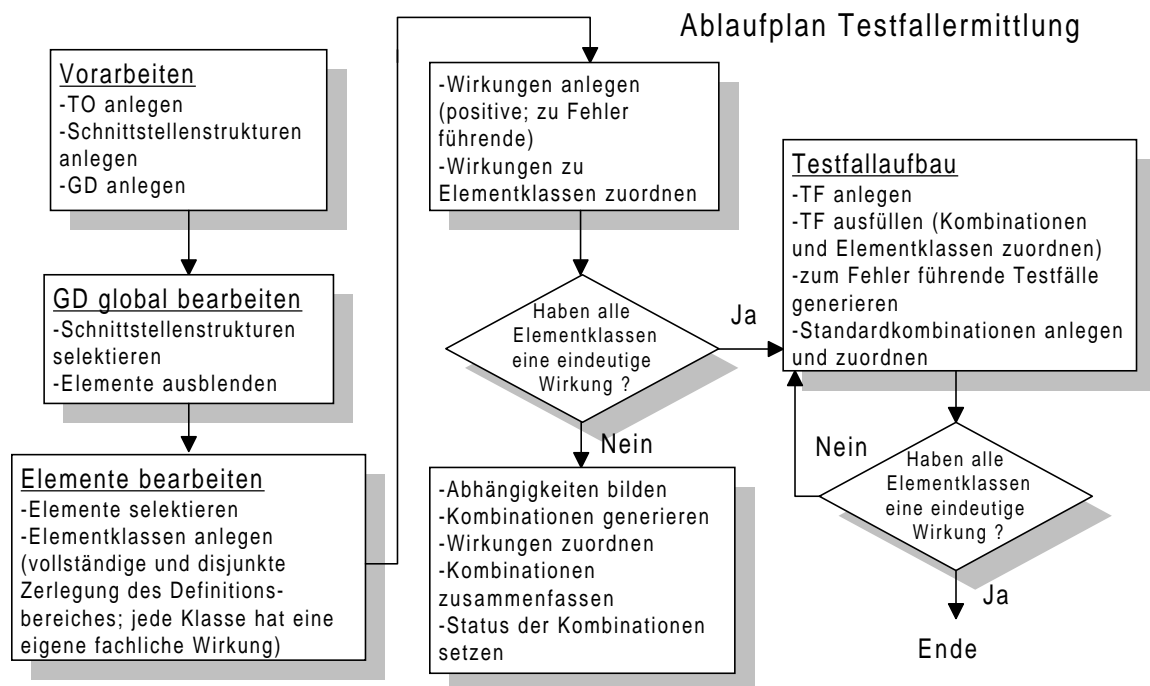


Abbildung 12.3-1: Ablaufplan zur Testfallermittlung

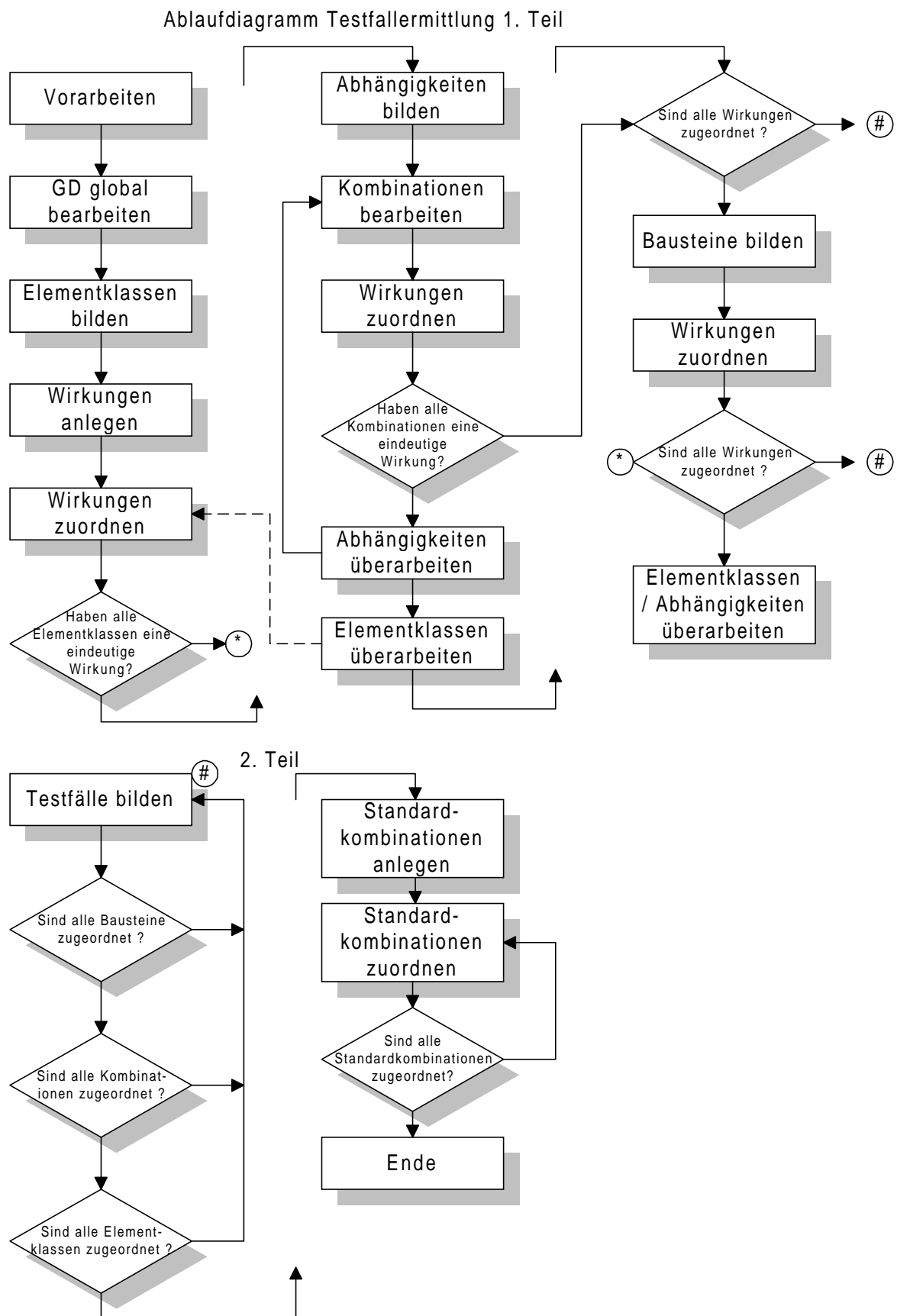


Abbildung 12.3-2: Ablaufdiagramm zur Testfallermittlung