

# INHALTSVERZEICHNIS

Vorwort ..... 24

## Kapitel 1: Eine Nudelmaschine für zu Hause

### Die SAP, das SAP-System und ABAP

Seite 27

Den Schrödinger hat es mal wieder erwischt: Ein österreichischer Nudelhersteller hat seine alte Firma, Spaghetti Infinito, übernommen. Dabei war das doch bis jetzt so gemütlich in der Spaghetti-Informatik. Man konnte programmieren, wie und womit man wollte – Hauptsache, es lief. Wenn er zukünftig keinen Spaghetticode mehr programmieren darf, kann er sich gleich etwas Neues suchen: Zum Beispiel bei der ABAP-Fabrik, von der er schon so viel gehört hat. Klingt lustig, ABAP. Und wenn die auch Code produzieren ..., vielleicht kann er dann ja doch wieder ein bisschen Spaghetticode programmieren! An seinem ersten Tag lernt er den Schwaiger Roland kennen, seinen ABAP-Ausbilder. Der seine Vorliebe für Spaghetticode sofort entdeckt und sich zunutze macht. Aber anders, als Schrödinger denkt.

Servus Schrödinger!!! .....	28	SAP-System .....	40
Schrödingers SAP-System – Eigenes .....		Die Erfolgsgeschichte: Vom ABAP zum ABAP .....	43
SAP-System runterladen .....	31	Kompilieren/Interpretieren .....	47
Schrödingers SAP-System – Das brauchst du dafür .....	33	ABAP allgemein .....	48
Schrödingers SAP-System – Installieren .....	34	ABAP-Syntax .....	49
Testsystem in der Wolke .....	37	Beispiel ABAP .....	51
Die SAP – Das Unternehmen .....	39	Schrödingers ABAP .....	52

## Kapitel 2: Rein in die Küche – raus aus der Küche

### Starten mit dem SAP-System

Seite 53

Ein spannender Tag für den Schrödinger: Das erste Mal! Der Schwaiger Roland meint, das tut gar nicht so weh, wie alle sagen, der hat leicht reden. Und fängt gleich mit der Anmeldung am SAP-System an. Äh, wo eigentlich? Ach ja, SAP Logon. Geschafft, und jetzt das Anmeldebild. Wie? Ach ja, SAP GUI. Aber was soll das mit diesen

Anmelden zum Tauchkurs: Eintauchen in die SAP-Welt .....	54	Die Auslage des SAP-Systems – SAP GUI-Aufbau .....	63
Alles ist Transaktion – Transaktionen als Shortcuts .....	56	Alles auf einen Blick – SAP Easy Access .....	65
Melde dich an! – SAP-Anmeldung .....	57	Modus Operandi – Zusätzliche Fenster öffnen ....	67
Transaktion starten .....	59	Hüben und drüben, nichts als Üben – Rolands Folterkammer oder Übung macht den Meister .....	69
Willkommen Schrödinger! – SAP-Willkommenstext .....	60		

## Kapitel 3: Ciao a tutti! (Hallo Welt!)

### Der 20-Minuten-Einstieg in die ABAP-Programmierung

#### Seite 71

Endlich, das erste ABAP-Programm! Schrödinger war sich bislang gar nicht sicher, ob der Schwaiger Roland überhaupt programmieren kann. Und dann legt er auf einmal richtig los: Pakete anlegen, Datenbankzugriffe, Ausnahmebehandlung, Schlüsselwordokumentation. Dieses ABAP fängt an, Spaß zu machen, weil es so viel zu entdecken gibt.

Einsteigen und anschnallen! .....	72	Kühlschrankdesign – DB-Tabelle anlegen .....	93
Object Navigator – Die integrierte Entwicklungsumgebung .....	74	Daten verarbeiten – Programm anlegen .....	97
Entwicklung organisieren – Systemlandschaft, Änderungsauftrag .....	75	Das ABAP-Einmaleins – Grundlegende Syntax ....	98
Ihr neuer Auftrag lautet: Auftragsanlage .....	77	Der Durchblicker – Programm entwickeln .....	100
Organisationskapsel Paket .....	79	Her mit den Daten – DB-Zugriff .....	102
Vom Entwickler zum Pakettier – Paket anlegen .....	81	Jetzt lese ich .....	103
Einen Schritt zurück und bald zwei nach vorne .....	84	Ohne Struktur läuft nichts – Grundlegende Programmstruktur .....	104
Die Infos zum Aufbau einer Datenbanktabelle ...	85	Alles muss es auch nicht sein – Einfaches Selektionsbild .....	106
Das zweischichtige Domänenkonzept .....	86	Layouten und Daten transportieren – Ein einfaches Dynpro .....	110
Technisches Fundament eines Typs – Domäne anlegen .....	88	Dekorieren der Auslage – Dynpro-Layout .....	112
Lege die Bedeutung an – Datenelement anlegen .....	91	Ablauflogik ohne Ablaufdatum – Ablauflogik programmieren .....	118
		Ein Shortcut für den User – Transaktionscode anlegen/ausführen .....	122

# Kapitel 4: AAA – Alles außer ABAP

## ABAP-Infrastruktur und -Werkzeuge

Seite 123

Der Schrödinger hat es schon gemerkt: Zum ABAPen reicht es nicht aus, sich die Sprache ABAP selbst reinzuziehen. Denn wo liegen die Programme, welche Programme gibt es eigentlich, womit werden sie bearbeitet? Welche Speicherbereiche werden genutzt? Wie kommen die Programme zu den Anwendern? Und ungefähr weitere 1.000 Fragen. Aber der Schwaiger Roland ist ja (noch) geduldig.

Warum? .....	124	Werkzeugkiste – Entwicklungswerkzeuge .....	139
Die Ablage der Entwicklungsobjekte –		Object Navigator – Der Regisseur .....	140
Repository .....	125	Repository Browser verschwinden lassen .....	141
Wer sucht der findet – Suchen mit dem		Repository Browser ausnutzen .....	143
Repository Infosystem .....	130	Repository Browser browsen .....	144
Suche von oben – Suchen in der		Synchron oder aus dem Tritt –	
Anwendungshierarchie .....	132	Objekt-Browser-Synchronisation .....	147
Suchen und Finden – Repository Infosystem .....	133	ABAP Editor – Die Schreibmaschine .....	149
Geschichtet, aber nicht gefaltet –		Debugger – Der Kammerjäger .....	152
ABAP-Infrastruktur .....	134	Debugger entschleunigt .....	153
Wohin mit dem Anwender –		Verwendungsnachweis –	
Speicherorganisation extern und intern .....	137	Der Rückwärtssucher .....	157

# Kapitel 5: Schräge Typen\*

## \*also jetzt nicht der Schwaiger Roland

## Datentypen und -objekte

Seite 159

Definieren und deklarieren: das tägliche Geschäft des ABAP-Programmierers. Das ist ja wie mit den unterschiedlichen Nudeltypen. Die eine Sorte ist lang und dünn, andere sehen aus wie Röhren oder wie Schmetterlinge. Jedoch, und das weiß der Schrödinger natürlich, wird für ein bestimmtes Gericht oder eine besondere Pastasoße ein bestimmter Nudeltyp benötigt. Und so ist es bei der Programmierung eben auch: Für bestimmte Aufgaben werden bestimmte Typen benötigt.

Vorspann .....	160	Variables Standgas – Datenobjekte .....	161
Von 0 auf 100 in wenigen Seiten – Technik der		Variablen sind zum Angreifen .....	162
Datenspeicherung im Hauptspeicher .....	160	Ein typischer Blickwinkel – Datentypen .....	163

Anders betrachtet – Datentypen .....	165	Her mit der internen Tabelle .....	185
Die Vermessung .....	166	Wie viel Spalten hätten Sie denn gerne –	
Ansprache – Datenobjekte verwenden .....	167	Zeilentyp .....	186
Dynamik pur oder Statik nur? – Statisch und		Normal, sortiert oder doch gehasht –	
dynamisch typisierte Variablen .....	168	Tabellenart .....	187
Mein lokaler Typ .....	169	Open the door please – Schlüssel .....	189
Zuerst mal elementar lokal – Lokale		Heute mal kopflos – Kopfzeile .....	190
elementare Typen .....	170	Tabellen für alle – Globaler Tabellentyp .....	191
Dann elementar global – Globale		Operation: Table – Tabellenoperationen .....	193
elementare Typen .....	172	Bibliothek einräumen .....	195
Elementar global con domäne – Domäne .....	175	Einzelne geht es auch – Einzelverarbeitung von	
Strukturell lokal – Lokale strukturierte Typen .....	178	Tabelleneinträgen .....	197
Struktur global – Globale strukturierte Typen .....	180	Ändern und Löschen zum Abrunden –	
Tabellarisch – typisch/intern –		Noch mehr Tabellenoperationen .....	198
Lokaler Tabellentyp .....	184		

## Kapitel 6: Si parla ABAP? Echtes Küchenlatein

### ABAP-Syntax I

Seite 199

Jetzt geht's ans Eingemachte, Schrödinger, oder sollten wir besser sagen ans Eingekochte? Hier lernst du mal das Basisvokabular, so wie damals, als nur Spaghetti gemacht wurden und nicht auch dieses andere Zeugs.

Von Kontrollen und Schleifen .....	200	Notandor – Logische Ausdrücke .....	214
Zuweisungen, Operationen, Funktionen .....	200	Verzweige, falls ... – IF ... THEN ... ELSE .....	216
Bitte nicht abweisen – Zuweisungen .....	203	Cassis – CASE ... WHEN ... ENDCASE .....	218
Was du darfst und was du sollst –		Schleifen .....	219
Konvertierungsregeln .....	205	Bedingungslose Schleifen – DO ... ENDDO .....	219
Ketten ohne Perlen –		Krasses Projekt für Hardcore-ABAPer .....	220
Zeichenketten-Funktionen .....	207	Bedingte Schleife – Fußgesteuert .....	222
Von Segmenten und Offsetten –		Bedingte Schleife – Kopfgesteuert .....	223
Zeichenketten-Operationen .....	211	Jetzt aber mal systematisch – Systemfelder .....	224
Ohne Rechnung, ohne mich – Arithmetische		Großbaustelle Rezeptprogramm .....	226
Operationen .....	213		

# Kapitel 7: Kleine Häppchen sind leichter wiederverdaut ( „Schluss mit Spaghetti!“ )

## ABAP-Syntax II

Seite 229

Schwaiger wirkt wild und entschlossen. Er hat endgültig genug vom Spaghetti-Coding, das Schrödinger fabriziert. „Heute mache ich aus dir einen Wiederverwender“, hallt die letzte Schwaiger-Drohung durch die Hallen, und Schrödinger versucht, sich mit Wehmut von seiner letzten Spaghetti zu trennen ...

Motivation durch Demotivieren .....	230	Ein Typ muss es sein – Schnittstelle typisieren ...	245
Den richtigen Zeitpunkt finden – Ereignisblöcke	231	Bitte mit Typ – Typisieren .....	246
Ereignisreiche Programme –		Warum in die Ferne schweifen? –	
Ereignisschlüsselwörter .....	232	Lokale Variablen .....	247
Ereignisse: Rekapitulation mit Selbstreflexion ....	233	Sichtbar oder nicht – Lokale Überdeckung .....	
Zur richtigen Zeit an irgendeinem Fleck? .....	235	Aufruf bitte – Unterprogramm-Aufruf .....	249
Meine schönsten Ereignisse –		Totalumbau .....	250
Rahmenprogramm .....	236	Globale Wiederverwendung –	
Dynpros mit Modul – PBO, PAI, HOV .....	238	Funktionsbausteine .....	254
Module, ganz kurz – Dynpro-Programmierung ...	239	Funktionsgruppe .....	256
Aber nun mit Schnittstelle – Unterprogramme ...	240	Leg dir eine Funktionsgruppe an .....	259
Definiere zuerst! Unterprogramm-Definition .....	241	Es ist so weit, global bereit –	
Übergeben und Übernehmen – Schnittstellen-		Funktionsbaustein .....	260
parameter für das Unterprogramm .....	242	Her mit dem Rezept .....	262
Lesen von DB mit Kapsel – DB-Zugriff in		Haaalloooooo – Funktionsbaustein-Aufruf .....	265
Unterprogramm .....	244	Dein Finale .....	267

# Kapitel 8: Schrödinger zeigt Klasse

## ABAP Objects

Seite 269

Schrödinger ist heute nicht gut aufgelegt, weil der Schwaiger Roland mit der Objektorientierung daherkommt. Es funktioniert doch alles bestens mit dem „prozeduralen ABAP“ (so nennt das der Schwaiger Roland). Was hat der bloß? Na okay, ein paar Mal haben sich schon Fehler eingeschlichen, die schwer zu finden waren. Mit der Objektorientierung soll ja alles besser werden mit der Ordnung. Und wo mehr Ordnung ist, ist alles leichter zu finden – sogar die Fehler. Und außerdem kann man mit ABAP Objects anscheinend tolle GUIs realisieren. Nach dem ersten Schockerlebnis ist Schrödinger also gleich überzeugt.

Motivation zur OO .....	270	Mit Methode – ran an den Code .....	302
Begriffe aus der Realität .....	271	Die Schnittstelle der Methode .....	306
Holistische Sichtweise .....	273	Ganz spezielle Methoden .....	309
Modellierung .....	274	Methoden mit Fülle .....	311
Erst denken, dann lenken: Einfache UML als		Selbstreferenz .....	314
Analysewerkzeug zur Visualisierung von		Statisches Attribut .....	315
Klassen/Objekten .....	278	Methodenaufruf .....	316
Klasse Kaffee(-vollautomat) auf dem Papier .....	280	Laufzeit sequenziert .....	318
Ohne meine Kollaborateure bin ich nichts .....	282	Methoden für den Wasserbehälter .....	320
Klasse oder doch Objekt .....	284	Von lokal nach global .....	322
Klasse Lokal .....	286	Going global – SE24 .....	323
Dein kleines Objektistan .....	289	Attribute und Methoden .....	324
Kaffee für zwei .....	292	Kaffeebohnen im Behälter .....	328
Datenablage – Attribute .....	294	Senden und Empfangen .....	332
Wasserstand und Pause .....	296	Sender mit Sendungsbedürfnis .....	333
Klasse Attribute, oder was? .....	297	Empfänger .....	334
Privat verkalkt/Private Attribute .....	300	Objektorientiertes Verkuppeln .....	335
Ab jetzt mit Methode – Schrödinger frischt auf ...	301	Kaffee s'il vous plaît .....	337

## Kapitel 9: Erben oder nicht, das ist hier die Frage: Pasta di Erbe

### Vererbung

Seite 339

Hoffentlich ist der Schrödinger heute in der passenden Stimmung. Eine Vertiefung der Objektorientierung, puh, da wird er schwitzen. „Was, da geht noch mehr? Jetzt hab ich mich gerade erst vom ersten OO-Schock erholt!“ Schrödinger wird wohl erkennen, dass Spaghetti, Ravioli, Tortiglioni etc. alles Nudeln sind, oder?

Motivation .....	340	Redefinieren – Polymorphie von Methoden .....	357
Vererben und Erben von Klassen .....	343	Redefinieren als globale Herausforderung .....	360
Vererbung lokal .....	346	Statische und dynamische Typen von Klassen	
Meister der Pyramiden .....	348	mit ein wenig Casting .....	366
Globale Vererbung:		Festigung der Einsichten .....	368
Ein praktisches Beispiel im SAP-Standard .....	350	Down-Cast .....	371
Was wird wie und wo vererbt .....	353	Abstrakte Klassen .....	374

# Kapitel 10: Keine Details bitte

## Der Blick durch Interfaces

Seite 379

„Man kann auch noch einen Schritt weiter gehen und die Implementierung von Methoden von deren Schnittstellendefinition trennen“, verkündet Roland und legt eine Kommunikationspause ein. Schrödinger steht da, der Mund ist offen, und er ist sich noch nicht der Tragweite der Aussage bewusst.

Schizophrenie? .....	380	Von Suchmaschinen und Tilden .....	389
Lokal zum Einsteigen .....	383	Hast du gerufen? .....	390
Schnittig und definiert .....	384	Die vielen Gestalten der Methoden .....	392
Volles Risiko – alles oder nichts .....	385	Globalisierung mit positiven Auswirkungen .....	395
Klasse Zuordnung .....	386	Global klasse Implementierung .....	397
Klasse Definition .....	387	Singleton-Party .....	399
Klasse Implementierung .....	388		

# Kapitel 11: Das Kapitel für Feiglinge

## Ausnahmen

Seite 403

„Fehlerbehandlung ist für Feiglinge! Das ist ja so, als ob ich beim Nudelkochen die Nudeln koste, ob sie al dente sind oder nicht!“ Schrödinger hält nichts von Ausnahmebehandlungen. Bis zum ersten Zwischenfall: als seine Freundin sich über klebrige und zerkochte Nudeln beschwerte.

Ausnahmsweise behandelt .....	404	Ausnahmen melden (mit Nummern) ... ..	416
Eine kleine Geschichte der Ausnahmen .....	406	... und behandeln (nach Nummern) .....	417
Nochmal langsam: Mit SY-SUBRC.		Ausnahmslos Objekte .....	419
Ausnahmewert wird gesetzt ... ..	410	Definieren geht vor .....	421
... und von dir behandelt .....	411	Ausnahmen melden (mit Ausnahmeklassen) ... ..	423
It's RAISING man, hallelujah. ....	412	... und Ausnahmen behandeln, aber richtig!	
Ausnahmen melden ... ..	412	Mit Ausnahmeklassen. ....	426
... und behandeln .....	413	Das ausnahmengespickte Projekt .....	427
The Catcher in the System .....	416		

# Kapitel 12: Spaghetti-Programmierung

## Selektionsbilder, Listen und Nachrichten (GUI I)

Seite 433

Es ist so einfach, eine SAP-Oberfläche zu programmieren: Ein Selektionsbild erstellen, eine Liste ausgeben, vielleicht noch ein paar Nachrichten an den Anwender verschicken – und fertig ist das berühmt-berüchtigte SAP GUI! Das klingt nach Spaghetticode und ist damit ganz nach dem Geschmack von Schrödinger. Zum Einstieg in die GUI-Programmierung ist es der beste Weg, einfach mal ein paar Elemente auf dem Bildschirm auszugeben. Für das richtige Ambiente sollen dabei Klettererdbeeren und Vogerlsalat sorgen ..., sagt der Schwaiger Roland.

Vom kleinen Window-Farmer bis zum Groß-GUI-Besitzer: Jeder will ein Selektionsbild! .....	434	Mehr, Mehrere, Mehreres (ohne Rauschen) .....	456
Selektionsbild für Beschränkte mit Liste .....	436	Zur Anwendung gebracht ... fast .....	460
PARAMETERS: Das kleinere der Selektionsbild-Ungeheuer .....	438	Graue Theorie: Eingabehilfe, Prüftabelle .....	462
Typ und Vorschlag .....	441	Entfessele den Künstler in dir: Screen-Gestaltung .....	464
Hey, Checker! .....	443	Endlich Texte! .....	468
Rund wie ein Radiobutton .....	445	Wie im Hamsterrad .....	470
„Du darfst“ war gestern, heute ist „Du musst“ ....	447	Ereignisreich geht's weiter .....	476
Abflug in den Keller .....	448	Tagesschau ... also eine Nachrichtensendung .....	479
Aber satt war er noch immer nicht: SELECT-OPTIONS .....	451	L – Li – Lis – List – Liste .....	483
		Keine Beichte notwendig: Interaktion mit einer Liste .....	490

# Kapitel 13: Lasagne aufs Auge

## Dynpro-Programmierung (GUI II)

Seite 493

„Wow, ein dynamisches Programm!“ Das gefällt Schrödinger sofort, dieses Dynpro. Dynamik ist einfach klasse, außer natürlich, es geht um Sport. Mit der Ablauflogik, Elementen zur Gestaltung der Darstellung und Eigenschaften, die man selbst programmieren kann: Super! Das ist schon richtig modular. (Ja, so langsam gewöhnt sich Schrödinger an die Feinheiten im ABAP-Vokabular.) Und es sind auch schon Schichten erkennbar, die Bausteine zur Dynpro-Programmierung sehen fast schon aus wie eine geschichtete MVC-Architektur. Und Schrödinger liebt Schichten, vor allem in Form von Lasagne!

Dynamisch programmieren .....	494	Alles ist im Fluss und manchmal geht es im Kreis .....	503
Wer schreit hier so? Rahmenprogramme, die Dynpros rufen! .....	498	Wohin soll ich mich wenden? .....	507
Dreieinigkeit .....	501	Wo Module bestimmen .....	508



Mit welchem Titel darf ich dich ansprechen? .....	511	Dein Dynpro .....	533
Über der Fläche steht die Oberfläche .....	514	Sammle Elemente .....	535
Die Gestaltung eines eigenen Menüs .....	517	Wohin mit den Daten? .....	537
Kannst du mit dem Druck umgehen? .....	520	Ablauflogik Reloaded .....	539
FFFFF .....	523	Dynpro rufen mal anders .....	547
Weck den Künstler in dir .....	525		

# Kapitel 14: Ravioli

## Web-Dynpro-ABAP-Programmierung (GUI III)

### Seite 551

Heute wartet ein besonderer Leckerbissen auf Schrödinger: Web Dynpro ABAP. Webanwendungen erstellen ohne HTML-Kenntnisse. Das klingt gut, findet Schrödinger. Ist aber für den ABAP-Programmierer eher ein i-Tüpfelchen. Kann man gut online lesen. Also: Schau auf der Bonus-Seite unter <http://www.sap-press.de/3593>.

# Kapitel 15: Raus aus meiner Küche!

## Berechtigungen

### Seite 553

„Hör mal, Schwaiger Roland, zu viele Köche verderben doch bekanntlich die Nudelsoße. Gibt es Möglichkeiten in ABAP, dass nicht jeder alles darf? Also dass ich bestimmten Anwendern manche Aktionen verbieten oder erlauben darf? Wenigstens einschränken?“ Der Schwaiger Roland lächelt nur weise.

Berechtigungsrundumschlag – Überblick		Der Berechtigungs-Selbstchecker –	
Berechtigungen .....	554	S_TCODE prüfen .....	562
Am Anfang steht das Objekt mit Klasse –		Experimente mit AUTHORITY-CHECK .....	564
Berechtigungsobjekt .....	557	Warum mag mich keiner – SU53 .....	565
Die Details zur Berechtigung .....	560		

# Kapitel 16: Vorratskammer einrichten mit ziemlich viel Schnickschnack

## DB-Tabellen erstellen

Seite 567

Was geschieht, wenn Anwender alle Daten im GUI eingegeben haben? Wohin gehen die Daten dann? Die müssen doch gespeichert werden? Denn die Anwender werden wahrscheinlich nicht immer wieder die gleichen Daten eingeben wollen – außer wenn sie an einem schwachen Kurzzeitgedächtnis leiden. Das würde sich Schrödinger öfter wünschen, das mit dem Kurzzeitgedächtnis, denn dann könnten sich die Anwender nicht an seine Programmierfehler erinnern ..., aber der Schwaiger Roland erklärt ihm dann doch lieber, wie er eine SAP-Datenbanktabelle erstellt.

Freiland-Datenhaltung – Daten persistieren .....	568	Ich will auch anders suchen – Sekundärindex .....	592
Warum einfach, wenn es mit Schnittstelle geht –		Ändern oder nicht, was geht –	
Die Datenbankschnittstelle .....	570	Erweiterungskategorie .....	595
Transparente Tabelle en Detail .....	573	Definieren und Realisieren –	
Spalten und der Rest – Tabellenfelder .....	576	Datenbankobjekt .....	597
Ohne Technik keine Tabelle – Technische		Mein erster Eintrag – Datenbanktabellen-	
Einstellungen .....	581	Einträge erzeugen .....	598
Mehr als eine Tabelle .....	584	Artenvielfalt im Dictionary –	
Welcher Wert ist möglich – Fremdschlüssel .....	587	Weitere Tabellenarten .....	600

# Kapitel 17: Vorratskammer plündern

## DB-Tabellen auslesen

Seite 601

Daten rein, Daten raus, und das möglichst einfach: So wünscht sich das der Schrödinger. Da kann ihm der Schwaiger Roland helfen. Und alleine die Begriffe „Open SQL“ und „ANSI SQL“ klingen wie Musik in seinen Ohren. Und er kann dann Daten aus einer oder sogar mehreren Datenbanktabellen lesen. Manchmal wünscht sich Schrödinger, er könnte den Schwaiger Roland in der Vorratskammer einsperren. Nur über Nacht.

Erster Takt – SQL .....	602	Open SQL grundiert – Basisbefehlssatz aus	
Zweiter Takt – SQL .....	602	Open SQL .....	605
Eingelagert und geplündert – Datenpufferung ....	603	Verklausuliert – SELECT-Klausel .....	607
Datenmanipulator nativ – DML mit Native SQL ...	604	Wohin damit – INTO target .....	610
Einfach definieren – Datendefinition mit		Woher – FROM source .....	611
dem ABAP Dictionary .....	604	Aber bitte nicht alles – WHERE-Klausel .....	613
Alles recht und schön – Berechtigungen und		Da geht noch mehr – Weitere Klauseln .....	614
Konsistenzprüfungen .....	604	Und jetzt alles kombiniert – SELECT-Beispiel .....	615

Mehr als eine Tabelle ist auch okay – Views .....	620	Zusammenfassung in Bildern –	
Schritt für Schritt zum View mit relationalen		View-Definition .....	634
Operatoren .....	624	Mit Views programmieren .....	636
Da will ich auch nicht alles –		Weil du so brav warst – Programmieren	
Selektionsbedingungen .....	633	mit Views .....	638

## Kapitel 18: Vorratskammer in Schuss halten

### DB-Daten pflegen

#### Seite 641

Daten pflegen – den Ausdruck mag Schrödinger. Und denkt an die Kräutersammlung auf seinem Fensterbrett. Da müssen manchmal auch neue Kräuter eingefügt werden, mal muss er das eine durch das andere austauschen. Und für das Löschen hat er auch schon eine leckere Soßenidee. Datenpflegeservice Schrödinger!

Lesen ist Silber, Schreiben ist Gold –		Und noch eins und noch eins und ... – INSERT ...	656
DELETE, UPDATE, MODIFY und INSERT .....	642	Massendaten einfügen .....	658
Sichere Daten – Transaktionskonzept .....	642	Ändern muss man auch ab und zu – UPDATE .....	661
Von einem Zustand zum nächsten –		Massendaten .....	662
Datenbank-LUW .....	644	Weg damit – DELETE .....	664
Bist du konsistent? .....	647	Kombianweisung – MODIFY .....	667
Aktionen bündeln – SAP-LUW .....	648	Halt, Sperre! – Sperrkonzept .....	669
Und jetzt auch noch transaktional –		Die fünf Gebote der performanten	
SAP-Transaktion .....	649	DB-Programmierung .....	670
Datenmanipulator-Entwicklungsrahmen –			
Das Programm für die Open-SQL-Anweisungen ...	651		

## Kapitel 19: Mit Schirm, Charme – und vielleicht noch einem Melonensorbet

### Daten in Dateien, Datenablage ohne DB

#### Seite 671

„Warst du schon einmal bei einem richtig guten Italiener?“ Schrödinger ahnt nichts Gutes. „Also so einem, wo du bereits am Eingang dem Kellner deine Garderobe abgeben kannst.“ – „Ja, das Prinzip der Garderobe ist mir bekannt, wieso?“ – „Na ja, wäre es nicht klasse, wenn die Anwender in ABAP auch Dateien einlesen oder herunterladen könnten – also an der Garderobe abgeben könnten?“ Au backe, das ist ja mal eine miese Metapher. Aber es ist vielleicht trotzdem wichtig zu wissen, wie man mit Daten umgeht, wenn man kein Datenbanksystem hat.

Daten ohne Datenbank .....	672	Download now! .....	682
GUI-Loads – Upload und Download .....	672	Einen hamma noch – Upload .....	685
Frontend Services .....	674	Daten auf dem Applikationsserver .....	687
Pfad ermitteln .....	677	OPEN House – OPEN DATASET .....	688
Download now, zumindest vorbereitet –		DatenTRANSFER – Daten schreiben .....	689
Download vorbereiten .....	679	READ DATASET – Daten lesen .....	691
Auch nett für Datenwiederverwendung –			
Clipboard .....	680		

# Kapitel 20: Täglich wechselnde Speisekarten

## Dynamische Programmierung

Seite 693

Morgens frische Lebensmittel auf dem Markt einkaufen, mittags daraus eine Speisekarte zaubern. Jeden Tag neu. Schrödinger bekommt Hunger, aber darum kann es dem Schwaiger Roland jetzt ja nicht gehen. „Was genau ...“ – „Wenn du erst während der Laufzeit eines Programms die Informationen erhältst, die für die Ausführung des Programms nötig sind, zum Beispiel den Namen einer Datenbanktabelle, dann ...“ – „Okay, okay, ein neues Konzept. Sag mir, wie es heißt, aber dann geht es in die Kantine.“

Dynamische Programmierung hat nichts mit		Dynamisches Feld .....	703
Beweglichkeit zu tun .....	694	Dynamischer Typ .....	703
Feldsymbole .....	694	Dynamische Komponente .....	704
Datenrefs .....	696	Dynamische Bedingung .....	704
RTTS = RTTI + RTTC .....	698	Dynamisches Unterprogramm .....	706
Überschrift finden .....	699	Programmerzeugung im Hauptspeicher .....	706
Dynamische Tokens .....	701	Programmerzeugung im Repository .....	707

# Anhang: Transportmanagementsystem einrichten

Seite 711

Entwickeln ist das eine, transportieren das andere. Deine schönen ABAP-Programme wollen natürlich zum Benutzer weiterreisen, das heißt vom Entwicklungs- in ein Folgesystem, zum Beispiel das Produktivsystem. Dazu benötigen sie ein Reisebüro, auch Transportmanagementsystem genannt.

Bildnachweis .....	710	Index .....	715
--------------------	-----	-------------	-----