

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

## 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 .....	<b>54</b>	Die Auslage des SAP-Systems – SAP GUI-Aufbau .....	<b>63</b>
Alles ist Transaktion – Transaktionen als Shortcuts .....	<b>56</b>	Alles auf einen Blick – SAP Easy Access .....	<b>65</b>
Melde dich an! – SAP-Anmeldung .....	<b>57</b>	Modus Operandi – Zusätzliche Fenster öffnen ....	<b>67</b>
Transaktion starten .....	<b>59</b>	Hüben und drüber, nichts als Üben – Rolands Folterkammer oder Übung macht den Meister .....	<b>69</b>
Willkommen Schrödinger! – SAP-Willkommenstext .....	<b>60</b>		

## Kapitel 3: Ciao a tutt! (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 Schwager Roland überhaupt programmieren kann. Und dann legt er auf einmal richtig los: Pakete anlegen, Datenbankzugriffe, Ausnahmebehandlung, Schlüsselwortdokumentation. Dieses ABAP fängt an, Spaß zu machen, weil es so viel zu entdecken gibt.

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

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

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

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

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

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

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

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

# 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 Schwager Roland.

Vom kleinen Window-Farmer bis zum Groß-	Mehr, Mehrere, Mehreres (ohne Rauschen) .....	456
GUI-Besitzer: Jeder will ein Selektionsbild! .....	Zur Anwendung gebracht ... fast .....	460
Selektionsbild für Beschränkte mit Liste .....	Graue Theorie: Eingabehilfe, Prüftabelle .....	462
PARAMETERS: Das kleinere der	Entfessle den Künstler in dir:	
Selektionsbild-Ungeheuer .....	Screen-Gestaltung .....	464
Typ und Vorschlag .....	Endlich Texte! .....	468
Hey, Checker! .....	Wie im Hamsterrad .....	470
Rund wie ein Radiobutton .....	Ereignisreich geht's weiter .....	476
„Du darfst“ war gestern, heute ist „Du musst“ ....	Tagesschau ... also eine Nachrichtensendung .....	479
Abflug in den Keller .....	L – Li – Lis – List – Liste .....	483
Aber satt war er noch immer nicht:	Keine Beichte notwendig:	
SELECT-OPTIONS .....	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	
Wer schreit hier so? Rahmenprogramme,		im Kreis .....	503
die Dynpros rufen! .....	498	Wohin soll ich mich wenden? .....	507
Dreieinigkeit .....	501	Wo Module bestimmen .....	508

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

## 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.

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

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

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

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

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

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

## 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 .....	<b>703</b>
Beweglichkeit zu tun .....	<b>694</b>	Dynamischer Typ .....	<b>703</b>
Feldsymbole .....	<b>694</b>	Dynamische Komponente .....	<b>704</b>
Datenrefs .....	<b>696</b>	Dynamische Bedingung .....	<b>704</b>
RTTS = RTTI + RTTC .....	<b>698</b>	Dynamisches Unterprogramm .....	<b>706</b>
Überschrift finden .....	<b>699</b>	Programmerzeugung im Hauptspeicher .....	<b>706</b>
Dynamische Tokens .....	<b>701</b>	Programmerzeugung im Repository .....	<b>707</b>

## 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.

<b>Bildnachweis .....</b>	<b>710</b>	<b>Index .....</b>	<b>715</b>
---------------------------	------------	--------------------	------------