

Norbert Kessel / Veit Wadewitz- Gupta SQLWindows32 - Eine Einführung

Copyright 1997 Verlag Dr. Kessel, Eifelweg 37, 53424 Remagen-Oberwinter  
Tel./Fax: 02228-493

Homepage des Verlages: [www.forstbuch.de](http://www.forstbuch.de)

Homepage von Gupta: [www.guptaworldwide.com](http://www.guptaworldwide.com)

Der Inhalt des Buches wurde mit großer Sorgfalt erarbeitet, dennoch kann keine Garantie für vollkommene Fehlerfreiheit übernommen werden.

Das Buch ist urheberrechtlich geschützt, alle Rechte sind vorbehalten. Kein Teil dieses Buches darf ohne schriftliches Einverständnis des Autors in irgendeiner Form reproduziert oder sonst weiterverarbeitet werden.

Es liegt ein weiteres Buch mit dem Titel „SQLBase-Einführung im gleichen Verlag vor. Weitere Informationen und Downloads auf der o.a. Homepage des Verlages.

Dr. Norbert Kessel / Dr. Veit Wadewitz

# Gupta SQLWindows32

Handbuch zur Einführung in die Programmierung von Datenbankanwendungen  
unter Windows 32 Bit

ISBN: 3935638-24-8

[www.forstbuch.de](http://www.forstbuch.de)



# Inhaltsverzeichnis

<b>1</b>	<b>EINFÜHRENDE TEIL</b>	<b>5</b>
1.1	HINWEISE ZUR PROGRAMMIERUNG VON DATENBANKANWENDUNGEN	8
1.2	EINFÜHRUNG IN GUPTA TEAM DEVELOPER	17
1.2.1	Generelles Arbeiten mit dem Gupta Team Developer	17
1.2.1.1	Der Startbildschirm von SQLWindows	18
1.2.1.2	Die Gliederung einer Gupta-Anwendung	30
1.2.1.3	Erste Schritte: vier Beispiele	32
1.2.1.31	Variationen zum Thema "hello world"	32
1.2.1.32	Anzeigen von Daten in einer Tabelle	35
1.2.1.33	Kurven: Business-Grafik	42
1.2.1.34	Erzeugen eigener Klassen	43
1.2.2	SAL, die Programmiersprache von SQLWindows	47
1.2.2.1	Datentypen	47
1.2.2.2	Variable	49
1.2.2.3	Konstanten	58
1.2.2.4	Statements	60
1.2.2.41	Set	60
1.2.2.42	If, Else, Else If	60
1.2.2.43	Select Case	62
1.2.2.44	Call	63
1.2.2.45	On	64
1.2.2.46	While, Loop	64
1.2.2.47	Return	66
1.2.2.5	Funktionen	70
1.2.2.6	Operatoren	73
1.2.2.7	Kommentare	74
1.2.3	Exkurs in die objektorientierte Programmierung	75
1.2.4	Objekte und deren Verwendung	77
1.2.4.1	Objekte der Standard-Klasse	78
1.2.4.11	Top Level Objekte	78
1.2.4.11.1	Form Window	80
1.2.4.11.2	MDI Window	87
1.2.4.11.3	Dialog Box	89
1.2.4.11.4	Table Window	90
1.2.4.12	Child Objekte	129
1.2.4.12.1	Background Text	132
1.2.4.12.2	Frame	133
1.2.4.12.3	Line	134
1.2.4.12.4	Picture	134
1.2.4.12.5	Data Field	139
1.2.4.12.6	Multiline Text	140
1.2.4.12.7	List Box	142
1.2.4.12.8	Combo Box	148
1.2.4.12.9	Radio Buttons	149
1.2.4.12.10	Option Buttons	151
1.2.4.12.11	Check Box	154
1.2.4.12.12	Push Buttons	155

---

	1.2.4.12.13	Scroll Bars	157
	1.2.4.13	Menu-Objekte	158
	1.2.4.14	Graphics	163
	1.2.4.15	Erzeugen von Anwendungen und Klassen mit dem Wizard	166
	1.2.5	Fehler	171
1.3		DRUCKEN	176
	1.3.1	Der Report-Generator	177
	1.3.2	Drucken von Reports	185
	1.3.2.1	Drucken von Daten aus einem Table Window	186
	1.3.2.2	Drucken von Daten, die zunächst gelesen werden müssen	191
	1.3.3	Kreuztabellen (Crosstabs)	195
	1.3.3.1	Kreuztabelle mit Report Builder	195
	1.3.3.2	Kreuztabelle selbst codiert	197
	1.3.4	Drucken von Grafiken	202
	1.3.5	Drucken von Bildern	207
1.4		SQLCONSOLE	210
1.5		SQLTALK	216
1.6		DATABASE EXPLORER	218
<b>2</b>		<b>BEISPIELANWENDUNG</b>	<b>220</b>
	2.1	ADRESSEN (KUNDEN / LIEFERANTEN)	221
	2.2	TÄTIGKEITEN FESTLEGEN	223
	2.3	AUFTRÄGE SCHREIBEN	224
	2.4	WEITERE MÖGLICHKEITEN	226
<b>3</b>		<b>ERWEITERUNG/VERTIEFUNG</b>	<b>229</b>
	3.1	DDE MIT EXCEL UND WINWORD	229
	3.2	DRAG AND DROP	233
	3.3	TOOLTIP PUSHBUTTONS	235
	3.4	SETZEN VON PROPERTIES MIT DEM COMPONENT DEVELOPMENT KIT (CDK)	236
	3.5	VERWENDEN ANDERER DATENBANKEN	241
	3.5.1	Lotus Notes	241
	3.5.2	ODBC	242
	3.5.2.1	ODBC mit dBASE-Dateien	243
	3.5.2.2	ODBC mit Access-Datenbanken	247
	3.6	DATENBANKZUGRIFF ÜBER OLE DB	250
	3.6.1	Zugriff auf Access mit OLEDB	252
	3.7	BARCODE	253
	3.8	DIE DATEI SQL.INI	256
<b>4</b>		<b>ACTIVEX UND COM</b>	<b>258</b>
	4.1	ALLGEMEINES	258
	4.2	ACTIVEX INTEGRATION IN SQLWINDOWS	259
	4.2.1	Einstieg: Der Acrobat Reader	259
	4.2.2	Statische Automatisierung am Beispiel von Excel	263
	4.2.2.1	Einbinden einer Type Library mit dem ActiveX Explorer	263
	4.2.2.2	Erstellen der Anwendung	268
	4.2.2.3	ActiveX Integration mit Word	272
	4.2.2.31	Erstellen eines neuen Dokuments	272
	4.2.2.32	Arbeiten mit Textmarken	276
	4.3	ERSTELLEN VON COM-ANWENDUNGEN MIT SQLWINDOWS	278

---

4.3.1	Erstellen der Server-Komponente .....	278
4.3.2	Erstellen der Client Komponente .....	285
<b>5</b>	<b>WEB ENTWICKLUNG MIT SQLWINDOWS .....</b>	<b>290</b>
5.1	GUPTA WEB QUICK-OBJECTS .....	290
5.1.1	Die leere Anwendung .....	291
5.1.2	Die Web Objekte .....	293
5.1.2.1	Entwicklung eines Web-Clients .....	295
5.1.2.11	Java Script Beispiel .....	298
5.1.2.2	Implementierung der Anwendung .....	299
5.1.2.3	Die WebXMLTable .....	301
5.1.2.4	Andere Web Clients .....	308
<b>6</b>	<b>ANHANG .....</b>	<b>309</b>
6.1	DATENBANK ISLAND .....	309
6.2	BEGRIFFE UND DEREN ERKLÄRUNG .....	312
6.3	FUNKTIONEN VON GUPTA SQLWINDOWS .....	317
6.3.1	Systemfunktionen .....	317
6.3.1.1	Bearbeitung von Array-Inhalten .....	317
6.3.1.2	Farben und Schriftarten .....	318
6.3.1.3	Konvertierung von Datentypen .....	318
6.3.1.4	Bearbeitung von Datumswerten .....	319
6.3.1.5	ActiveX Funktionen .....	319
6.3.1.6	Debugging .....	320
6.3.1.7	Dialog Boxes .....	320
6.3.1.8	Zugriff auf DOS-Dateien .....	321
6.3.1.9	Drag- und Drop .....	322
6.3.1.10	Edit-Funktionen (Cut und Paste) .....	322
6.3.1.11	File Management .....	323
6.3.1.12	Einstellung von Formaten .....	324
6.3.1.13	List Boxes und Combo Boxes .....	325
6.3.1.14	Windows-Management .....	326
6.3.1.15	Background-Text .....	327
6.3.1.16	MDI (Multiple Document Interface) Windows .....	328
6.3.1.17	Versenden von Nachrichten .....	328
6.3.1.18	Diverse Funktionen .....	328
6.3.1.19	Numerische Funktionen .....	330
6.3.1.20	Bilder .....	331
6.3.1.21	Druck .....	332
6.3.1.22	Report-Bearbeitung .....	332
6.3.1.23	Scroll-Bar .....	333
6.3.1.24	SQL .....	333
6.3.1.25	Zeichenketten .....	334
6.3.1.26	Table Windows .....	335
6.4	NACHRICHTEN .....	338
<b>7</b>	<b>STICHWORTVERZEICHNIS .....</b>	<b>342</b>

## Vorwort zur zweiten, veränderten und erweiterten Auflage

Ziel dieses Buches ist es, zu beschreiben, wie Datenbankanwendungen mit dem Programmpaket Gupta SQLWindows entwickelt werden, die unter Windows 32-Bit Betriebssystemen ablauffähig sind. Es ist ein einführendes Buch zur Programmierung und richtet sich daher zunächst an den Anfänger, kann aber durch die Art der Strukturierung und den vielen Beispielen auch etwas für den fortgeschrittenen Programmierer bieten.

Großen Wert wurde darauf gelegt, die Beispiele isoliert und vollständig vorzustellen. So soll gewährleistet werden, dass die Beispiele leicht in eigenen kleinen Programmen ausprobiert werden können. Allerdings können nicht alle Funktionalitäten vorgestellt werden; die Auswahl der Themen ist naturgemäß subjektiv, sollte jedoch einen motivierenden Einstieg in die Programmierung von Datenbankanwendungen mit diesem leistungsfähigen Werkzeug ermöglichen.

Die vorliegende Ausgabe geht auf die Version 2.1 und auch 3.0 (zum Zeitpunkt der Erstellung Beta 4) des Gupta Team Developer ein. Die Funktionalitäten, die in früheren Releases (1.x) nicht oder anders implementiert sind, werden an der jeweiligen Stelle im Buch kenntlich gemacht.

Da die Basisfunktionalität des Gupta Team Developer sich seit der letzten Auflage nicht wesentlich geändert hat, wurden die einführenden Kapitel 1 und 2, mit Ausnahme des neuen Abschnitts 1.2.3 Exkurs in die objektorientierte Programmierung, im Wesentlichen unverändert aus der bisherigen Ausgabe übernommen.

Neu aufgenommen sind Abschnitte über das Component Development Kit, Tooltip Pushbuttons und den WebDeveloper, die auch schon in der Version 1.x des CTD zur Verfügung stehen.

Ebenso wird auf die OLE DB-Datenbankanbindung, ActiveX Integration – insbesondere Excel und Word -, Erstellung von COM Servern und Clients sowie die Web XMLTabelle eingegangen, die von Gupta ab der Version 2.x des Team Developer zur Verfügung stehen. Speziell diese Funktionalitäten ermöglichen es mit SQLWindows zeitgemäße und performante Anwendungen zu erstellen.

Die im Buch verwendete Beispieldatenbanken können von der Homepage des Verlages ([www.forstbuch.de](http://www.forstbuch.de)) heruntergeladen werden (s. dort: Download).

Der Firma Gupta (insbesondere Hn. Martin Teetz) möchten wir danken.

Lauf und Oberwinter im Oktober 2002

Veit Wadewitz und Norbert Kessel



# Kapitel 1

## Einführender Teil

### Einige Anmerkungen zu den Verzeichnissen auf der Festplatte

Bei der vollständigen Installation von Gupta unter Windows 32Bit wird eine Reihe von Verzeichnissen angelegt, deren Inhalte im folgenden kurz dargestellt werden.

Im Laufwerk, das für die Installation ausgewählt wurde, werden folgende Verzeichnisse angelegt:

- **\Gupta (ab Version 3.0) oder \Centura** Hierin sind die zur Programmierung von Datenbank Anwendungen nötigen Daten gespeichert.
- **\Sql80:** Falls bei der Installation angegeben wurde, dass die lokale SQLBase installiert werden soll, werden hierin die dafür nötigen Dateien abgelegt (alternativ könnte auch mit einer Datenbank auf einem Netzwerkverzeichnis gearbeitet werden, dann entfällt dieses Verzeichnis).

Die folgende Übersicht beinhaltet die in den beiden genannten Verzeichnissen enthaltenen Unterverzeichnisse kurz und die Extensionen der darin enthaltenen Dateien.

Übersicht der Unterverzeichnisse:

Verzeichnis	Unterverzeichnis	Dateiarten (Extensionen)	
<b>\Gupta 2.1</b>	\AXLibs	ActiveX APLs	
	\Deploy	Exe, Wse, Ini	
	\Docs	HTM	
	\Inc	H	
	\Island	Dbc, Log	
	\Samples	Apl, App, Bmp, C, Csv, Def, Dll, Exe, Ico, Ifx, Ini, Mak, Ora, Qrp, Reg, Sql, Wts, Xls, Cqt, Cpp, Dsp, Dsw, H	
	\Sysproc	Sql	
	\Templates	App	
	\Tools	Apt, exe	
	\Tutorial	App, Dll, Exe, Tlb, Ico, Gif	
	\uninstall	Exe	
	<b>SqlBase 8.0</b>	\Alarms	Fil
		\Books	Pdf, HTML
\include		Inl, H	
\island		Dbc, Log	
\Java		Class, HTM	
\lib		Lib, Dll	
\Redist		Setup.exe	
\relnotes_files		Release notes (Htm, Jpg)	
\samples		Sql, C, Dll, Java, Dsp, Dsw, Cpp, H, Rc, Mak, Exe, Vbp	
\scripts		Sql	
\src		Cpp, Dsp, Dsw, H, Rc, Rc2	
\Topic		HTM	

## Das Verzeichnis DEPLOY

Durch die Installation von Gupta, wird ein Verzeichnis mit dem Namen DEPLOY errichtet. In diesem Verzeichnis finden sich Dateien der Extensionen WSE, INI und eine Exe-Datei (deploy21.Exe bei SQLwindows 2.1).

Die Dateien stellen die Umgebung her, die benötigt wird, eine Anwendung auf einem Rechner ablaufen zu lassen. Das heißt, nachdem mittels Gupta ein Anwendungsprogramm geschrieben wurde, wird diese Anwendung als EXE-Datei gespeichert (in Gupta über den Menüpunkt Project/Build Settings - Exe). Dem Kunden müssen aber neben dieser Exe-Datei und eventuell der DBS-Datei mit der Datenbank – sowie ICO bzw. BMP Files, alle diejenigen Dateien ausgeliefert werden, die in dem Verzeichnis DEPLOY enthalten sind. Insgesamt ergibt sich also folgender Umfang bei Auslieferung eines Programms, das mit dem Team Developer entwickelt wurde:

- EXE-Datei, erstellt mit Gupta Team Developer;
- Alle Dateien im Verzeichnis DEPLOY;
- DBS-Datei(en) mit den darin enthaltenen Daten;
- SQLBase (z.B. die SQLBase-Runtime oder eine andere SQLBase, die der Kunde erwirbt).

**Tip:** Wie vorstehend erwähnt, benötigen sämtliche Anwendungen, die auf einem Fremd-Rechner installiert werden sollen, die im Verzeichnis Deploy enthaltenen Dateien. Hierzu bieten sich zwei Möglichkeiten an: Die CD mit der kompletten Entwicklungsumgebung wird zu dem Kunden mitgenommen und die Dateien unter Verwendung des Installationsprogramms installiert. Alternativ wäre es sinnvoll, die in dem Verzeichnis Deploy enthaltenen Dateien (in der vorliegenden Version sind es 3 Dateien mit einem Umfang von rund 17 MB) mit der Anwendung auf eine CD zu brennen.

## Wichtige Programme, die außerdem installiert werden

Bei der Installation von Gupta werden neben der Entwicklungsumgebung einige weitere Programme installiert: Die folgende Übersicht faßt die Anwendungen, deren Programm-Namen und eine kurze Beschreibung zusammen.

Name	Dateiname	Beschreibung
Gupta SQLWindows 2.1/3.0	Cbi21/30.Exe	Entwicklungsumgebung
Repository Setup Wizard 2.1/3.0	Tmdbi21/30.Exe	Installiert die Datenbank für das Repository (verwendet bei Team-Entwicklung)
Sample Application Explorer	Sampler.Exe	Beispielanwendung
SQLBase Database Engine 32bit	Dbnt1sv.exe	Datenbank-Maschine
SQLConsole DBA Utility 7.6/8.0	Sqlcon76/80.Exe	Programm zur Pflege von Datenbank-Dateien
SQLTalk Interactive SQL	Sqltalk.Exe	Dialog-Programm für SQL-Anweisungen
Team Developer Help 2.1/3.0	Centura/Gupta.Hlp	Hilfdatei für Entwicklungsumgebung
Team Object Manager 2.1/3.0	Tmi21/30.Exe	Zusammen mit der Repository-Datenbank für Verwalten von Projekten bzw. deren Versionen

Der Begriff "Gupta SQLWindows" wird in der Literatur und in der Online-Dokumentation zusammen mit dem Begriff "Gupta Team Developer" benutzt, auch in diesem Buch wird so verfahren.

Einige der in der vorstehenden Übersicht aufgeführten Programme und Dateien sollen im folgenden kurz vorgestellt werden.

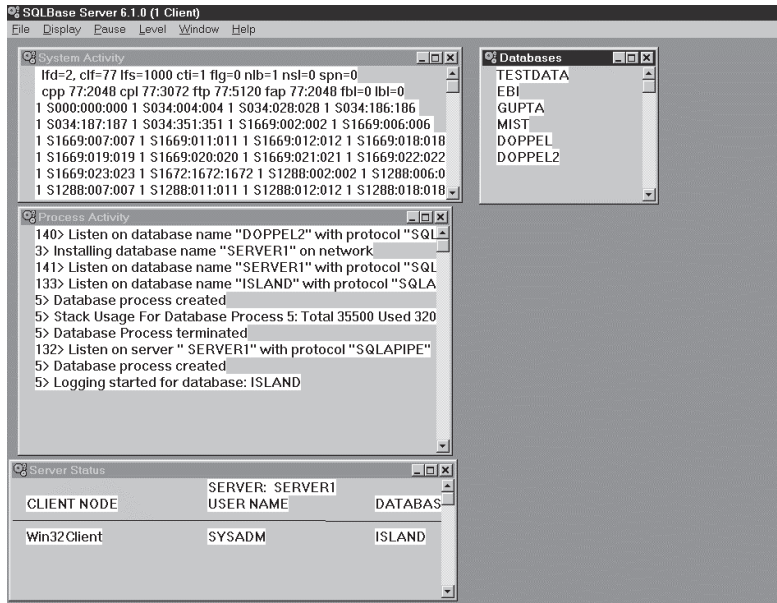
## SQLBase 32 Bit Server

Üblicherweise enthält eine selbst erstellte Anwendung eine Verbindung zu einer (oder mehreren) Datenbanken. Somit liegt also üblicherweise ein System vor, das aus zwei Programmen besteht: zum einen die eigentliche Anwendung, die vom Programmierer erstellt wurde, zum anderen die Datenbankmaschine, die Daten bereitstellt, die in den DBS-Dateien gespeichert werden. SQL-Anweisungen, die in der Anwendung enthalten sind, werden an die Datenbankmaschine geschickt, dort übersetzt, ausgewertet und eventuelle Ergebnismengen oder Meldungen zurück zur Anwendung „gesendet“.

Der SQLBase 32 Bit Server ist die Datenbankmaschine, die Daten bereitstellt und eingehende SQL-Anweisungen empfängt, ausführt und die Ergebnismengen an die Anwendung zurücksendet. Wird in Team Developer die Funktion **SqlConnect()** verwendet, dann wird ggf. die Datenbank-Engine gestartet (oder die nötigen Router bzw. ODBC-Treiber), ansonsten wird eine Verbindung zu ihr hergestellt (hierzu werden die unten erläuterten SQL Handles benötigt).

Nachdem die Datenbankmaschine gestartet wurde, zeigt sich nach einem Klick auf das Symbol (angezeigt in der Statuszeile) folgendes Bild.

Abb.: Datenbank-Maschine mit den diversen Fenstern mit Informationen (die Fenster werden über den Menüpunkt Display/All geöffnet).



Die einzelnen Fenster geben Informationen zu gerade geöffneten Datenbanken sowie zur Auslastung des Systems.

## Dateien, die bei der Arbeit mit Gupta Team Developer entstehen (können)

Im Grunde können bei der Arbeit mit dem Team Developer vier Arten von Dateien entstehen, Programmdateien, Bibliotheken (Libraries) oder COM-Server.

- Programmdateien können dabei in verschiedenen Formaten gespeichert werden, als **APP**-Datei (in normaler oder kompilierter Form) oder als **APT**-Datei (als normaler oder eingerückter Text).
- Bibliotheksdateien enthalten Funktionen, die mit Team Developer geschrieben wurden und die in einer separaten Datei gespeichert werden um sie für andere Programme verwendbar zu machen. Um diese Funktionen den anderen Programmdateien verfügbar zu machen genügt dann eine Anweisung, die diese APL-Datei oder APD-Datei einbindet. Wie das funktioniert, wird unten beschrieben.
- COM-Server werden in kompilierter Form als **EXE** oder **DLL** (lokal) oder als **MTS-DLL** (**M**icrosoft **T**ransaction **S**erver) erstellt

## Literatur, die zu Beginn gelesen werden sollte

Neben diesem Buch, das eine vollständige Einführung in die Programmierung von Datenbank-anwendungen bietet, sind im Lieferumfang von Gupta einige Online Bücher und Dokumente enthalten, die unbedingt gelesen werden sollten.

Die folgende Übersicht nennt einige der wichtigsten Dokumente.

Titel	Art	Wo zu finden	Inhalt
Introducing Gupta	Datei	Books Online	Einführung in Gupta
Developing with Gupta SQLWindows	Datei	Books Online	Allgemeine Hinweise zur Entwicklung von Anwendungen
Reporting with Gupta SQLWindows	Datei	Books Online	Programmierung von Reports
SQLConsole Guide	Datei	Books Online	Einführung in SQLConsole
SQLBase SQLTalk Command Reference	Datei	Books Online	Einführung in SQLTalk
Developing: Chapter 7 - SAL (Scalable Application Language)	Datei	Books Online (über Button mit Fernglas)	SAL-Funktionen
Developing: Chapter 17 - Libraries	Datei	Books Online	Bibliotheken
Extending: Chapter 11 - Visual Toolchest	Datei	Books Online	Weiterführende Funktionen (s.a. die Datei VTTEST.APP im Verzeichnis Samples)

## Wie dieses Buch aufgebaut ist

Dieses Buch gliedert sich in die folgenden Kapitel:

**Kapitel 1** führt in die Programmierung ein. Neben Hinweisen zur Verwendung der Sprachen SQL (Structured Query Language) und SAL (Scalable Application Language), werden die in einer Anwendung verwendbaren Objekte detailliert beschrieben. Neben vielen Beispielen zur Verwendung von Table Windows sind hier auch umfangreiche Anmerkungen zum Drucken von Daten aller Art enthalten.

**Kapitel 2** stellt eine Beispielanwendung vor, die über das Gupta-Forum von CompuServe ohne Gebühren heruntergeladen werden kann.

**Kapitel 3** bringt einige weitere Aspekte für die Programmierung, so z.B. Hinweise zu DDE, zur Arbeit im Netz, zur Verwendung anderer Datenbanken, zu Barcodes, Tooltips und dem CDK.

**Kapitel 4** befasst sich mit der ActiveX Integration und der Erstellung von COM Servern und Clients.

**Kapitel 5** zeigt die Entwicklung von Web Anwendungen mit SQLWindows

**Kapitel 6** enthält Übersichten mit Funktionen und Nachrichten, die inhaltlich zusammengefasst sind.

**Kapitel 7** enthält das Stichwortverzeichnis.

## Wichtige Begriffe

Einige Begriffe sind so grundlegend, dass sie gleich hier kurz erläutert werden sollen. Ausführliche und umfangreichere Begriffserläuterungen finden sich in den späteren Abschnitten.

**Front-End, Back-End, Client-Server:** Als Front-End wird eine Anwendung bezeichnet, die zusammen mit einem Back-End (s. unten) Kommandos oder Daten austauscht; beides zusammen wird auch als Client-Server-System bezeichnet. Hier ist unter Front-End eine Anwendung zu verstehen, die mit Gupta Team Developer erstellt wurde, allerdings sollte das durchaus erweitert betrachtet werden, da grundsätzlich (und grundsätzlich heißt, dass es Ausnahmen gibt) jede Windows-Anwendung als Client bzw. als Server auftreten kann. Beispielsweise könnte Winword oder Excel (genaugenommen ein darin geöffnetes Dokument) als Client für Daten verwendet werden, die aus einer Datenbank-Datei kommen, die mittels SQLBase geöffnet wurde.

**Datenbank oder Datenbank-Maschine:** Ein Programm, das Daten verwaltet, die in einer DBS-Datei gespeichert sind. Das heißt, neue Datensätze werden durch die Datenbank-Maschine in eine Tabelle eingefügt, zu löschende Datensätze werden durch sie gelöscht, Ergebnismengen, die durch Abfragen entstehen, werden durch die Datenbank-Maschine erzeugt und zu dem Front-End geschickt, von der die Anweisung kam. Datenbanken enthalten neben Tabellen, auch Indexe und einige andere Objekte, die üblicherweise mit dem Programm SQLConsole erzeugt oder gelöscht (allerdings ist es auch möglich innerhalb der Entwicklungsumgebung von Gupta Team Developer eine Datenbank anzulegen: Menüpunkt Database/Database Explorer).

**Tabelle:** Daten wie z.B. Kundendaten werden in Form sog. Tabellen gespeichert. Jede Tabelle hat einen eindeutigen Namen und ist definiert durch eine Struktur. Tabellen werden durch den Anwender unter Verwendung des Programms SQLConsole oder SQLTalk angelegt, verändert oder gelöscht; sie enthalten bis zu  $2^{15}$  Datensätze. (Anmerkung: Es ist auch möglich, Tabellen innerhalb des Team Developers anzulegen usw., allerdings sollte dies nur mit Test-Daten geschehen, da es besser ist, hierfür SQLConsole zu verwenden). Der Begriff Tabelle wird in zwei Formen verwendet, zum einen für Tabellen, die Daten beinhalten, die in einer Datenbank gespeichert werden, zum anderen für sog. Tabellen-Objekte, die in einer Anwendung eingebettet sind, und in denen z.B. Daten aus einer Datenbank-Tabelle dargestellt werden. Neben den Tabellen, die vom Anwender angelegt werden, sind in einer Datenbank immer solche Tabellen enthalten, die für den Betrieb des Systems wichtig sind, so z. B. eine Tabelle mit dem Namen Systables, in der alle Tabellen aufgeführt sind oder eine Tabelle Syscolumns, in der alle Spalten (aller Tabellen) aufgeführt sind. Diese Tabellen können verwendet werden, wenn eine Übersicht der Tabellen oder der Tabellen-Strukturen gedruckt werden soll.

**Index:** Ein Index ist eine Hilfstabelle, die es der Datenbank-Maschine ermöglicht, schneller Daten auszuwählen, zu suchen oder zu sortieren. Bei der Auswahl von einigen wenigen Datensätzen aus einer großen Menge verringert sich die Antwortzeit üblicherweise drastisch, so dass eine allgemeine Empfehlung für Indexe ausgesprochen werden kann, auch wenn das Eingeben oder Löschen von Daten etwas länger dauert.

**Abfrage:** Als Abfrage bezeichnet man eine Anweisung, die aus einer gegebenen Menge von Datensätzen eine Teilmenge auswählt, die eine bestimmte Eigenschaft erfüllt, beispielsweise aus einer Menge von Aufträgen diejenigen, die noch nicht bezahlt sind.

**Datentyp:** Datentypen legen fest, welche Art von Information in einer Variablen oder der Spalte einer Tabelle zu speichern sind. Wichtig ist, dass Datentypen bei der Codierung festgelegt werden müssen und bei der Struktur von Tabellen (gelegentlich müssen Datentypen bevor sie gespeichert werden können, zuerst umgeformt werden, dies wird unten erläutert).

**Normalisierung:** Durch die Normalisierung werden Redundanzen abgebaut, das heißt Daten werden so gespeichert, dass beispielsweise ein Kunden-Datensatz nur einmal vorkommt und dass alle Rechnungen, die für diesen Kunden erstellt werden über die Kundennummer diesem einen Kunden zugeordnet werden können.

**SQL** (Structured Query Language): SQL ist eine Sprache, die verwendet werden kann, Datenbanken, Tabellen, Datensätze, Indexe usw. anzulegen, zu ändern oder zu löschen. Diese Sprache wird typischerweise verwendet um Datensätze zu verarbeiten.

**SAL** (Scalable Application Language): SAL ist die Sprache, die in Gupta Team Developer verwendet wird, um Aktionen, die mit Objekten auf dem Bildschirm stattfinden sollen, zu codieren. Beispielsweise wird eine Funktion mit dem Namen SalTblPopulate() dazu verwendet, ein Tabellenobjekt in einer Anwendung mit den Datensätzen einer Datenbank-Tabelle zu füllen. In Gupta Team Developer sind über 400 solcher Funktionen enthalten.

# Kapitel 1.1

## Hinweise zur Programmierung von Datenbankanwendungen

### Allgemeines

Eine Datenbankanwendung besteht aus einem Programm (einer Exe-Datei), die mittels Gupta Team Developer erstellt wurde und die Objekte enthält (z.B. ein Tabellen-Objekt, Datenfelder usw.). Um Daten aus einer Datenbank-Tabelle anzuzeigen, müssen SQL-Anweisungen unter Verwendung der Sprache SAL an die Datenbank-Maschine geschickt werden, die diese dann auswertet.

### Die verwendeten Daten

Für die Einführung in die Programmierung werden in diesem Buch die Daten verwendet, die durch die Installation auf die Festplatte übertragen wurden. Sie befinden sich in der Datenbank mit dem Namen **Island.Dbs** (\Sqlbase\Island\Island.dbs). Dies soll einen schnellen Einstieg gewährleisten. Nachfolgende Abschnitte beschreiben einige der Tabellen, die in dieser Datenbank enthalten sind und für dieses Buch verwendet werden (im Anhang des Buches findet sich eine Übersicht aller Tabellen, die in der Datenbank Island.Dbs enthalten sind).

### SQL

In Gupta werden zwei Sprachen nebeneinander verwendet, SAL (Scalable Application Language) und SQL (Structured Query Language). Einige grundlegende Anmerkungen zu der Sprache SQL folgen hier.

Die Datenbanksprache SQL wird für alle Zugriffe auf die Datenbank verwendet. Demgegenüber wird für die Ablaufsteuerung von Anwendungen (das heißt wenn ein Fenster geöffnet wird, ein Bericht gedruckt wird usw.) die Sprache SAL verwendet. Manchmal sind Elemente beider Sprachen auch gemischt zu verwenden, dann nämlich, wenn eine SAL-Funktion aufgerufen wird, die mit Daten aus Tabellen arbeitet. Später wird als Beispiel hierfür die Funktion `SalTblPopulate()` vorgestellt, eine SAL-Funktion, die zum Füllen einer Tabelle mit Datensätzen aus einer Datenbank verwendet wird, und in deren Klammern u.a. eine SQL-Anweisung enthalten sein muss.

Hier sollen nur einige wenige SQL-Anweisungen vorgestellt werden, die es ermöglichen, Datensätze in eine Tabelle einzugeben, sie zu ändern und sie zu löschen. Somit soll gewährleistet werden, dass auch der SQL-Lai schnell mit der Programmierung von Datenbankanwendungen beginnen kann.



---

 Einige SQL-Anweisungen mit Erläuterungen

Anweisung	Erläuterungen
select Kundennummer, Name, Vorname from Kunden	Lese aus der Tabelle Kunden die Inhalte der Spalten Kundennummer, Name, Vorname und zeige sie an.
select, Name, Vorname from Kunden where Kundennummer = 10000	Lese den Inhalt der Spalten Name und Vorname aus dem Datensatz, dessen Kundennummer 10000 ist.
delete from rechnungen where status = 2	Lösche alle Datensätze aus der Tabelle Rechnungen, deren Inhalt im Feld Status 2 ist.
update artikel set vkpreis = vkpreis * 1.10	Ersetze in der Tabelle Artikel den Inhalt des Feldes vkpreis mit dem Wert von vkpreis * 1.10, das heißt die Preise aller Artikel werden um 10 % erhöht.
select max from kundennummer + 1	Ermittle die größte Kundennummer und addiere die Ziffer 1 dazu. Dies wird häufig verwendet, um eine neue (numerische) Kundennummer zu erzeugen

---

Die Anweisungen können (und sollten) mit einem Front-End-Programm, wie z.B. SQLTalk, SQLConsole oder dem Database Explorer innerhalb von Gupta getestet werden.

## Libraries (Bibliotheken)

Libraries sind Dateien, die mit Funktionen gefüllt sind. Diese Funktionen (die unter Verwendung von SAL oder aber auch C) geschrieben sein können sind häufig das Ergebnis von umfangreichen Anwendungen, in denen immer wieder bestimmte Aktionen nötig sind, so beispielsweise das Öffnen einer Datenbank, das Löschen von Daten in Formularen oder das Lesen von Datensätzen aus einer Tabelle. Nachdem der Programmcode für diese Aktionen so aufgebaut wurde, dass die Funktion unter Verwendung von Variablen bzw. Parametern aufgerufen werden können, werden die Funktionen in einer separaten APL-Datei gespeichert. Diese APL-Datei enthält - neben den Funktionen - keine anderen Teile wie z.B. Formulare oder Table-Windows. Durch das Einbinden der APL-Datei in eine eigene Anwendung (die typischerweise die Endung APP oder APT hat) werden alle Funktionen für diese Anwendung verfügbar. Es ist auch möglich, die Datei Newapp.App (s. unten) mit diesem Eintrag zu versehen, so dass die Bibliotheken in jeder neuen Anwendung zur Verfügung stehen. Der Sourcecode eingebundener Bibliotheken wird in der Hauptanwendung standardmäßig in Blau dargeseht und ist dort nicht editierbar.

Ein weiterer Typ von Bibliotheken sind die Dynamic Libraries (APD). Sie enthalten neben Funktionalität auch Graphische Komponenten und sind eigene „kleine“ Anwendungen. APDs werden zur Laufzeit dynamisch eingebunden und können mit der aufrufenden Anwendung kommunizieren. Auf diese Weise ist es möglich, Programmfenster zwischen mehreren Anwendungen zu teilen, ohne dass der volle Sourcecode in jeder Anwendung enthalten sein muss. APDs dienen also hauptsächlich dazu, Clientressourcen zu schonen. Der funktionale Code der APD ist in der Hauptanwendung nicht sichtbar, in der APD muss der String `!__Exported` hinter der Deklaration von Top-Level Objekten und Functions gesetzt werden, um diese Objekte in der Hauptanwendung zu sehen und ansprechen zu können.

## Classes (Klassen)

Eine Class ist eine Vorlage, die die Eigenschaften von Objekten festlegt. Beispielsweise kann eine Table-Class bereits mit Variablen so ausgestattet sein, dass später nur noch die Inhalte dieser Variablen festzulegen sind. Von Anfang an dabei ist die sog. Standard-Klasse, in der Eigenschaften der Objekte (wie z.B. von Data Fields) festgelegt sind. Wird ein Data Field in ein Formular eingefügt, dann ist der Datentyp festgelegt auf „String“, das heißt, dieses eben eingefügte Data Field erwartet String-Daten (numerische Daten können dagegen nicht dargestellt werden und führen zu einer Fehlermeldung). Natürlich werden oft Data Fields benötigt, in denen numerische Daten dargestellt werden müssen. Anstatt jetzt jedesmal den Typ von String auf Number umzustellen und die Ausrichtung vom linken auf den rechten Rand und das Format festzulegen (das heißt die Anzahl der Nachkommastellen), wird man eine neue Klasse erzeugen, in der die eben genannten Eigenschaften einmal festgelegt werden. Fügt man danach ein neues Data Field in ein Formular ein, dann kann zwischen der Standard-Klasse und der selbst erzeugten Klasse gewählt werden, nachträgliches Ändern entfällt damit.

## Objekte (Objects)

Ein Objekt einer Anwendung ist ein Push Button, ein Table-Window usw., also Teile der Anwendung, die mit Programmcode versehen werden können. Objekte verarbeiten außerdem Nachrichten, die beim Auftreten von Ereignissen (z.B. eines Mausklicks) verarbeitet werden können. Man unterscheidet Standard-Objekte, Quick-Objekte und selbst definierte Objekte, die jeweils in Klassen zusammengefasst werden.

Generell existieren zwei Typen von Objekten, **Top-Level** oder Parent-Objekte, die als Container für **Child Objekte** dienen. Top-Level Objekte können Parameter, Variablen, Funktionen, ggf. Menüs und Aktionen enthalten und sind zur Laufzeit in ihrer Größe veränderbar. Child-Objekte enthalten nur Aktionen. Eine Ausnahmestellung hat nur das Child-Table Window, auf das später näher eingegangen wird.

Typischerweise beginnt man in Gupta damit, die Objekte der Standard-Klasse zu verwenden um kleinere oder auch größere Anwendungen zu erstellen. Spätestens dann aber, wenn zum zehnten Mal ein Push Button mit der Anweisung zum Minimieren eines Form Windows oder zum Beenden einer Anwendung codiert wird, stellt sich die Frage, ob das nicht einfacher ginge. Es geht, allerdings ist es gerade am Anfang kompliziert, weil man hierzu abstrahieren und das Gemeinsame (z.B. aller Push Buttons zum Minimieren eines Fensters) erkennen muss.

Mit zunehmender Erfahrung aber wird man feststellen, was bei ähnlichen Modulen das Gemeinsame (bzw. das nicht-Gemeinsame) ist. Das Gemeinsame lässt sich dann in Form einer Klasse bzw. eines Objekts dann so allgemein codieren, dass es sich in allen zukünftigen Anwendungen einbauen lässt.

## Nachrichten (Messages):

Nachrichten werden innerhalb von SAL-Anwendungen verschickt, um über bestimmte Programmschritte oder vorgenommene Aktionen zu informieren. So kann unter Verwendung der Nachricht "SAM\_Click" auf einfache Weise Bezug zwischen einem Push Button und der

Aktion des Anwenders (nämlich dem Klicken auf eben diesen Push Button) genommen werden. Neben den vorgefertigten Nachrichten, die mit der Zeichenfolge "SAM" beginnen (eine Auflistung findet sich im Anhang) besteht auch die Möglichkeit, eigene Nachrichten zu erzeugen und zu verwalten. Die Verwendung dieser Nachrichten, die z.B. mit der Zeichenfolge "AM" beginnen könnten (das ist beliebig wählbar, sollte jedoch immer unterscheidbar sein von SAM-Nachrichten), erlaubt es, flexiblere Programme zu schreiben.

Dieser Sachverhalt hat für den Programmierer große Bedeutung: Programme unter Windows können dem Anwender nun nicht mehr starr vorgeben, was er tun oder lassen kann, so wie das mit normalen DOS-Programmen möglich und auch normal war. Jetzt muss ein Programm so erstellt werden, dass jederzeit auf jede mögliche Eingabe eines Anwenders reagiert werden kann. Der Programmierer muss im voraus dafür sorgen, dass alle möglichen Eingaben des Anwenders richtig aufgenommen und verarbeitet werden.

Während dies bei Verwendung von Programmiersprachen wie z.B. "C" oder "Pascal" nicht unbedingt einfach ist, ist Team Developer dafür schon vorbereitet, das heißt die Fähigkeit zur Reaktion auf vielfältige Eingaben ist bereits in ihm enthalten. Der Programmierer prüft z.B. nur noch, ob der Anwender in einem zuvor aktivierten Fenster den OK Push Button gedrückt hat. Da der dann auszuführende Programmcode an diesen Push Button angebunden ist, kann die Aktion des Anwenders korrekt weiter verarbeitet werden.

Ausgelöst werden Messages durch Ereignisse (Events). Diese können von der Hardware (Mausklick, Tastatur), aus dem Betriebssystem (z.B. Timer) oder aus der Anwendung (z.B. Überschreitung eines gesetzten Grenzwertes) ausgelöst werden.

Drei Arten von Nachrichten werden in Gupta Team Developer unterschieden (weitere Hinweise finden sich unter dem Menüpunkt Help/Messages in Team Developer:

- Nachrichten mit der Vorsilbe SAM, die im System schon vordefiniert sind,
- Nachrichten mit der Vorsilbe WM, die von Microsoft Windows vordefiniert sind, und
- eigene, selbst zu definierende Nachrichten, die z.B. mit AM beginnen (AM ist nicht vorgegeben, wird aber üblicherweise verwendet).

## Beispiele zu SAM-Nachrichten:

<b>SAM-Nachricht</b>	<b>Bedeutung/wann gesendet</b>
SAM_AppStartup	Start der Anwendung
SAM_App_Exit	Beenden der Anwendung
SAM_SqlError	Fehler bei einer Sql-Anweisung
SAM_Click	Drücken der linken Maustaste (entspricht WM_LBUTTONDOWN)
SAM_Create	Nachdem ein Objekt erzeugt wurde
SAM_SetFocus	Ein Objekt erhält den Fokus
SAM_Destroy	Ein Objekt wird zerstört (vom Bildschirm entfernt)
SAM_Report*	Mehrere Nachrichten für den Druck von Reports
SAM_DDE*	Mehrere Nachrichten für DDE-Protokoll

Die Nachrichten können nicht beliebig verwendet werden. Beispielsweise können in der Sektion Application Actions einer Anwendung (das heißt ganz oben) nur drei Nachrichten verwendet werden (SAM\_AppStartup, SAM\_AppExit, SAM\_SqlError). Dagegen kann die Nachricht

SAM\_Create bei fast jedem Objekt eingesetzt werden, das heißt, man kann für ein Form Window ebenso wie für einen Push Button die Nachricht einsetzen und verarbeiten.

Anmerkung: Ist die Visual Toolchest Library in eine Anwendung eingebunden, dann stehen zusätzliche Nachrichten zur Verfügung, so z.B. VTM\_RightClick, VTM\_MiddleClick aber auch VTM\_RightDoubleClick usw. Dies Nachrichten werden genauso wie die SAM\*-Nachrichten verarbeitet und erhöhen die Flexibilität von Anwendungen deutlich.

## Beispiele zu WM-Nachrichten:

Nachrichten mit der Vorsilbe WM stammen von Microsoft Windows, sie sind in der Datei Windows.H definiert und können dort eingesehen werden (die Datei ist im Lieferumfang des MS-Windows SDK bzw. in MS C++ enthalten).

<b>SAM-Nachricht</b>	<b>Bedeutung/wann gesendet</b>
WM_LBUTTONDOWN	Linke Maustaste wurde gedrückt
WM_LBUTTONUP	Linke Maustaste wurde losgelassen
WM_RBUTTONDOWN	Rechte Maustaste wurde gedrückt
WM_DDE*	Mehrere Nachrichten für DDE

## Beispiele zu AM-Nachrichten:

Diese Nachrichten werden durch den Programmierer festgelegt; typischerweise wird deren großer Nutzen erst nach einer gewissen Erfahrung erkannt. Die nachstehende Übersicht kann deshalb nicht „vollständig“ sein und soll hier nur als Anregung dienen.

<b>AM-Nachricht</b>	<b>Bedeutung/wann gesendet</b>
AM_Emp_Select	Wenn die Tabelle für die Darstellung der Employee-Daten gefüllt wird
AM_Inv_Select	dito, für die Tabelle Invoice (bei Tabellen in der Datenbank Island.Dbs enthalten)

Um eine solche Nachricht zu deklarieren, ist in der Sektion Constants/User einer Programm-Datei folgender Eintrag nötig:

```

Constants
    System
    User
  
```

**Number: AM\_Emp\_Select = SAM\_User + 1**

Nachrichten sind immer vom Typ Number (da intern hierfür eine Nummer verwendet wird). Neben der Festlegung dieses Typs muss der Name der Nachricht eingetragen werden (Groß- und Kleinschreibung ist wie immer zu beachten). Dann folgt die System-Konstante SAM\_User, sie steht für die zuletzt definierte Nachricht im System, zu der die Ziffer 1 addiert wird, um eine neue, noch freie Nummer zu erhalten. Analog könnte für eine weitere Nachricht mit SAM\_User + 2 eine andere freie Nummer ermittelt werden. Dieses Vorgehen entlastet von dem Bemühen, eine noch freie Zahl zu finden, außerdem ist eine so codierte Nummer auch in neuen Pro-