



FISCHER & CONSULTANTS

## Schulungs-Zertifikat

über das F&C-Training On The Job

## Sybase SQL Anywhere Advanced Database Developer

ausgestellt für

<Teilnehmer>

<Firma>

<Teilnehmer> hat das Training im Umfang von 80 Stunden mit dem umseitig aufgeführten Curriculum erfolgreich abgeschlossen und seine Kenntnisse gegenüber dem F&C-Trainer <Trainer> im Praxisaudit bewiesen.

Dortmund, den <Datum> ,

FISCHER & CONSULTANTS  
GESELLSCHAFT FÜR UNTERNEHMENSBERATUNG  
UND SOFTWAREENTWICKLUNG MBH  
MARTINSTRASSE 1  
44137 DORTMUND  
TEL 0231 – 14 36 34  
FAX 0231 – 14 36 46  
INFO@APPFAC.T.DE

n



Computer Associates™  
Authorized Training Partner  
CA-Visual Objects



FISCHER & CONSULTANTS

# Sybase SQL Anywhere

## Advanced Database Developer

### Curriculum

**Grundbegriffe:** Client/Server- und Mehrschichten-Architekturen ♦ Funktionale Datenzugriffs-APIs (ODBC) und objektorientierte APIs (ADO und JDBC) ♦ Datenbank-Objekte (Tabellen, Spalten, Indizes, Constraints, Trigger, Prozeduren, Views, Klassen) ♦ Metadaten und Systemtabellen ♦ Administratorwerkzeuge Sybase Central, ISQL und Debugger ♦ Case-Tools ♦ Grundzüge von SQL: Data Definition Language (DDL) und Data Manipulation Language (DML).

**Database Design/Creation:** Einführung Sybase PowerDesigner ♦ Grundlagen konzeptioneller E/R-Modelle (CDM) ♦ Entities, Attributes, Primary Keys, Alternate Keys, Domains, Constraints und Checks ♦ SQL Datentypen ♦ Relationen (1:1, 1:n, n:m) und Foreign Keys ♦ Dependend Relations, attributierte und nicht-attributierte n:m-Relationen ♦ Vererbte Tabellen ♦ Grundlagen physischer E/R-Modelle (PDM) ♦ Owner und Tablespaces ♦ Indexdefinition ♦ Regeln der deklarativen referentiellen Integrität: Restrict, Cascade, Set Null, Set Default bei Updated und Delete ♦ Script-Generierung aus dem E/R-Modell und DDL ♦ Datenbank-Initialisierung und wichtige Datenbank-Parameter ♦ Einspielen des Datenmodells per ISQL ♦ Change-Management mit Hilfe von CASE-Tools.

**Data Manipulation Language:** Einfache SELECTs ♦ Spaltenauswahl und Virtuelle Spalten ♦ Casting und Formatierung ♦ Skalare Funktionen (Length, Left, Right, Case etc.) ♦ Rechengenauigkeit bei numerischen Spalten ♦ Grundlagen des Filterns mit WHERE ♦ Fortgeschrittene Filter und Subselects mit IN, EXISTS und ANY ♦ Sortieren mit ORDER BY ♦ Aggregatfunktionen, GROUP BY und HAVING ♦ Verknüpfungen und kartesisches Produkt ♦ JOINS und OUTER JOINS ♦ Subselects als Alternative zu JOINS ♦ Kontrolle des Query Optimizers mit grafischen Access Plans ♦ VIEWS ♦ Grundlagen von INSERT, UPDATE und DELETE ♦ INSERT auf Basis von SELECT ♦ UPDATE und DELETE über JOINS ♦ Transaktionen, COMMIT und ROLLBACK ♦ Isolation Level und Locking.

**SQL-Programmierung:** Überblick: Funktionen, Prozeduren und Trigger ♦ Parameterübergabe in Funktionen und Prozeduren ♦ Variablen in SQL ♦ Rückgabewerte von Funktionen und Prozeduren ♦ Kontroll-Statements (IF, CASE, LOOP, WHILE) ♦ Cursor-Programmierung und Cursor-Typen ♦ Exception-Handling und User-Defined-Errors ♦ Prozeduren mit Result-Sets ♦ Trigger-Typen, -Zeitpunkte und Row-Aliase ♦ Fehlen des

Mutating Table Syndroms bei Sybase T-SQL geg. PL/SQL ♦ Verfahren zur Primary-Key-Generierung ♦ Extended Procedures für Programmaufrufe, Loggen von Windows-Ereignissen und Email aus SQL ♦ SQL-Entwicklungs-Zyklus, Debugging und Remote Debugging.

**Client-Programmierung:** Vergleich ODBC, ADO und JDBC ♦ Objektmodell von ADO im Überblick (Connection, Command, Parameter, Recordset, Field, Error) ♦ Connection Strings für den Anywhere OLEDB-Provider ♦ Ausführen von Commands und Verwendung von Parametern ♦ Arbeiten mit Recordsets, Cursortypen und Lockings ♦ Exception-Handling und die Error-Klasse ♦ Transaktions-Programmierung ♦ Objektmodell von JDBC im Überblick (DriverManager & Connection, Statement, ResultSet) ♦ Registrierung des Sybase Drivers und Connection URLs für Anywhere ♦ Statements und die Verwendung von IN-, OUT- und Stream-Parametern ♦ Arbeiten mit ResultSets und Datentypen ♦ SQLException und SQLWarning ♦ Transaktions-Programmierung.

**Java in the Database (JitD):** Überblick: Datenbank als JVM ♦ Aufruf von Java-Klassen aus SQL-Statements ♦ Laden eigener Klassen und Packages in die Datenbank ♦ SQL-Zugriff aus JitD-Methoden (JDBC und Internal Connections) ♦ Wrappen von JitD-Static Methods durch SQL Stored Procedures (Signaturen von Wrapper-Prozeduren) ♦ Typischer Entwicklungszyklus von JitD-Code (Entwicklung außerhalb der Datenbank, Final Testing als JitD) ♦ Tuning-Tips: Caching der Default-Connection, Collections im Java-Heap als intelligenter Cache.