

Detail-Weiterbildungskonzept Digital Factory

No- & Low-Code Entwickler

Allgemeines

Das folgende Detail-Konzept konkretisiert die im Dokument *03_Konzeption_Digital Factory_NoLowCode.pdf* vom Juni 2021 enthaltenen Inhalte. Zielgruppe und Vermittlungsansatz sind dort ebenfalls beschrieben.

Präzisierungen:

- Zur Auflockerung und Vertiefung der Themen sind Übungs- und Diskussionseinheiten vorgesehen. Die Übungen können von den Teilnehmenden an Hand vorbereiteter Referenzlösung nachvollzogen werden. Eigene Implementierungen "from scratch" sind nicht vorgesehen, Interessierte können jedoch selbstverständlich eigeninitiativ Erweiterungen einbringen.
- Alle Ressourcen werden den Teilnehmenden in elektronischer Form bereitgestellt. Hierzu ist ein GitHub-Repository vorgesehen, das auch nach Ende der Sessions verfügbar bleibt.
 - Zur Benutzung des Repositories sind keine Kenntnisse von Git/GitHub erforderlich!
- Eine Unterrichtseinheit dauert 90 Minuten, 4 Einheiten pro Tag ergeben eine Netto-Dauer von 6 Stunden.
 - Der mittlere Anteil von Übungen/Diskussionen eines einzelnen Blocks ist etwa 25%.

Einführung Datenbanken und Datenbankmanagement (4 Tage)

Hinweis

Als Beispiel für ein Datenbank-System wird MSSQL verwendet. Ein Datenbank-Server wird zentral zur Verfügung gestellt.

Schulungsinhalte

- Einführung (2 Unterrichtseinheiten)
 - Aufgaben einer Datenbank (Zentrale Datenhaltung, Konsistenz, Integrität)
 - Kategorisierung (Schwerpunkt Relationale Datenbank-Systeme, Kurzer Überblick in die Welt von NoSQL)
 - Produkt-Übersicht
 - Erster Kontakt mit der MySQL-Datenbank, Schwerpunkt Datenbank-Software wie DBeaver, MySQL-Workbench, PHP-Admin, SQL-Developer
- Aufbau einer einfachen Tabelle (1 Unterrichtseinheit)
 - Organisation von Daten in einer tabellarischen Struktur
 - Spaltentypen
- Basics der Abfragesprache SQL (1 Unterrichtseinheit)
 - SELECT-Statement
 - WHERE-Klausel
- Aufbau einer einfachen Datenbankstruktur (2 Unterrichtseinheiten)
 - Entity-Relationship-Model und ER-Diagramme
 - Primär- und Foreign Keys, Auto-Increment
 - Werkzeuge zur Modellierung
- Die Abfragesprache SQL (2 Unterrichtseinheiten)
 - Joins
 - Berechnungen und Analysen
- Aktualisierung von Datensätzen (2 Unterrichtseinheiten)
 - Notwendigkeit und Formulierung von Constraints im Datenmodell
 - INSERT, UPDATE und DELETE
 - Datenkonsistenz durch Transaktionen (ACID)
- Ein Überblick in die Themen der Datenbank-Administration (2 Unterrichtseinheiten)
 - Spot the Error, Finden und Lösen von DB-Problemen
 - Datenbanken upgraden
 - Datenbanken organisieren (Rollen & Rechte)
 - Staging
 - Views
- Software-Architekturen (1 Unterrichtseinheiten)
 - Wie nutzen Applikationen/Frontend-Anwendungen eine relationale Datenbank?
 - Mehrschichtige Anwendungen
- Ein komplettes Beispiel (3 Unterrichtseinheiten)
 - Beispiel und Analyse eines komplexen Datenmodells
 - Diskussion von Möglichkeiten und Auswirkungen von Änderungen im Datenmodell
 - Praxisbeispiele

Datenbank Software (3 Tage)

Hinweise

- Als Beispiel für ein Datenbank-System wird MSSQL verwendet. Ein Datenbank-Server wird zentral zur Verfügung gestellt.
- Als Beispiele für NoSQL-Datenbanken werden Couchbase und Apache Cassandra benutzt. Auch hier werden die Datenbank-Instanzen zentral bereitgestellt.

Schulungsinhalte

- Die Datenbank als Applikationsserver (4 Unterrichtseinheiten)
 - Indexing
 - Stored Procedures
 - Database Extensions, Beispiel MSSQL
 - Trigger
- Datenbank-Administration (1 Unterrichtseinheit)
 - Staging
 - Migration
 - Datenbanken Performance (Abfragepläne, Performanceoptimierung)
 - Tablespace
- Business Intelligence (1 Unterrichtseinheit)
 - Strategien und Technologien zur effizienten Datenanalyse
 - Data Warehouses und Data Marts
- Das Internet als Datenbank (2 Unterrichtseinheiten)
 - RESTful Webprojekte
 - CRUD-Services
 - JSON als Standard-Datenformat
- NoSQL (2 Unterrichtseinheiten)
 - Das ER-Modell revisited
 - Not Only SQL: Alternative Modellierung und Typisierung
 - Vertiefung Dokumenten-orientierte Datenbanken, Dokumente versus Pages
 - Entscheidungskriterien für die Auswahl einer NoSQL-Datenbank
 - Setup einer NoSQL-Datenbank
- Big & Fast Data (2 Unterrichtseinheiten)
 - Probleme größerer Datenmengen, Skalierung / Clustering von Datenbank-Instanzen
 - NoSQL versus SQL

Einführung Software Entwicklung (4 Tage)

Hinweise

- Dies ist kein Programmier-Seminar, Programm-Codes werden bereitgestellt
- Das Full-Stack-Beispiel zeigt ein Angular-Frontend mit einem C#-Webservice und Datenablage in einer MSSQL-Datenbank

Schulungsinhalte

- Was ist Software? (1 Unterrichtseinheit)
- Software Sprachen (3 Unterrichtseinheiten)
 - Skript- und Compiler-Sprachen
 - Prozedurale, Objekt-orientiert, Funktional
 - Beispiele (C#/Java, Python, PHP, JavaScript)
- Rollen und Anforderungen (3 Unterrichtseinheiten)
 - Aufstellen eines Entwickler-Teams
 - Technische Anforderungen
 - Verfahren und Werkzeuge zur Team-Kommunikation
 - Die Werkzeug-Palette der Software-Entwicklung
- Software-Lifecycle (3 Unterrichtseinheiten)
 - Von der Planung zum Betrieb
 - Die Bedeutung von Test und Qualitätsmanagement
 - Erweiterung durch neue Features
- Services (3 Unterrichtseinheiten)
 - Was sind Services?
 - Service-Dokumentation und Schnittstellenmanagement
 - Automatisierte Bereitstellung von Software mit CI/CD
- Präsentation der Entwicklung einer Full-Stack-Anwendung (3 Unterrichtseinheit)