|  |  |  |
| --- | --- | --- |
| **Kurs:** Databases | | **Code:** W4DATABAS |
| **Studiengang** | Bachelor Wirtschaftsinformatik | |
| **Studiengruppe** | BWI-A21 | |
| **Dozent/in** | Walter Rothlin | |
| **Leistungsnachweis(e)** | 2 schriftliche Arbeiten (Gewichtung je 50%) | |
| **Präsenz** | **Selbststudium** | **Workload** |
| 32 h | 58 h | 90 h |

|  |
| --- |
| **Lernergebnisse** |

Die Studierenden

* kennen den generellen Aufbau, die Funktion und den Einsatz von Datenbanken
* entwerfen aufgrund von Geschäftsanforderungen ein logisches Datenmodell
* implementieren das physische Datenmodell in der DB (inkl. Views, Functions and Stored-Procedures)
* können mit Hilfe von SQL Anweisungen Abfragen in der Datenbank ausführen
* können von Python aus eine DB Verbindung via Connection-String aufbauen, eine SQL Abfrage starten und das Resultat anzeigen

|  |
| --- |
| **Lehrinhalte** |

Die Studierenden

* können ein ERD lesen und erweitern, sowie ein Neues anhand von Requirements erstellen
* können in einem bestehenden Schema (sakila) Abfragen (inkl. Joins) in SQL formulieren
* können CRUD Operationen in einer bestehenden DB formulieren und ausführen
* können Functions, Views und Stored-Procedures erstellen
* können eine eigenes Schema entwerfen und DB-Objekte mit SQL (DDL) erstellen, erweitern, ändern und löschen
* von Python aus auf ein Schema zugreifen und Abfragen auslösen

|  |
| --- |
| **Lehr- und Lernmethoden** |

* Stoffvermittlung Frontalunterricht, Übungen lösen im Präsenzunterricht und Selbststudium
* **Formative Lernkontrolle:** Basierend auf dem Referenz-Schema (sakila) im DBMS System MySQL müssen die Studierenden zur Vorbereitung auf den Leistungsnachweis Fragenstellungen implementieren und die Resultate selbständig überprüfen.

|  |
| --- |
| **Leistungsnachweise** |

2 schriftliche, handlungsorientierte Leistungsnachweise via Moodle, Einzelarbeiten, open book

Gewichtung: je 50%

Zeitbegrenzung:

1.Leistungsnachweis: 90’

2.Leistungsnachweis: 3h

|  |
| --- |
| **Lehrmittel** |

Empfohlen

* Moodle-Kurs mit Links zu Online-Tutorials

|  |  |
| --- | --- |
| **Thema Block 1: ERD** | |
| Datum | Di 21.3.2023 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Installation MySQL inkl sakila Schema

Inhalte (Folien 1-20)

* Reverse-Engineering von sakila mit der Workbench
* DDL und DML Aktionen mit Workbench via GUI ausführen
* Datentype
* Primary key (Unique, Not Null)
* ERD-Symbolik

Nachbereitung

* Eigenes durch ***Reverse Engineering*** produziertes ERD von sakila ergänzen

|  |  |
| --- | --- |
| **Thema Block 2: CRUD DML (Select, Insert, Update, Delete)** | |
| Datum | Di 4.4.2023 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Folien 21 – 29 studieren und Fragen notieren

Inhalte

* Fragen beantworten / erklären
* INSERT, UPDATE, DELETE mit der Workbench
* SQL Scripts ausführen (sakila drop and create, Load data)
* SELECT
  + AS Alias
  + DISTINCT
  + WHERE (= und !=, LIKE, NOT NULL, NOT IN, BETWEEN, REGEXP BINARY, AND, OR)
  + Functions-Aufrufe (DATE\_FORMAT)
  + ORDER BY
  + GROUP BY

Nachbereitung

* Aufgaben lösen (Select, Functions, Meta-Data)

|  |  |
| --- | --- |
| **Thema Block 3: Normalisierung / Inner Joins / Outer Joins / Views** | |
| Datum | Di 11.4.2023 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Was ist normalisieren und welche Konsequenzen hat das auf SQL-Abfragen?

Inhalte (Folien 30 – 38)

* Normalisierung
* Joins
  + Inner Joins
  + Outer (Left) Joins
* Views

Nachbereitung

* Aufgaben lösen (Joins, SubQueries)
* Select mit Inner-Join erstellen (Adressliste der Angestellten)
* Eigene Person als Mitarbeiter mit Workbench erfassen und SQL analysieren
* 3 weitere Sprachen in sakila definieren und bei 10 Filmen die Originalsprache setzen (via Scripts und Workbench) und wieder Rückgängig machen

|  |  |
| --- | --- |
| **Thema Block 4: Transaktionen, Functions, Stored-Procedures** | |
| Datum | Do 18.4.2023 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Was ist eine Transaktion und wann ist das zwingend erforderlich?
* Was sind die Software-Engineering Überlegungen hinter Functions and Procedures in einer DB?
* Wie werden Functions und Proceduren in MySQL geschrieben, getestet und verwendet.

Inhalte (Folien 39 – 44)

* Transaktionen
* Eigene Functions definieren und aufrufen
* Eigene Proceduren definieren und aufrufen

Nachbereitung

* Übung-Beispiele nachprogrammieren und austesten

|  |  |
| --- | --- |
| **Thema Block 5: 1. Leistungsnachweis** | |
| Datum | Di 9.5.2023 |
| Zeit | **17:30 – 20:45** |

Inhalte

* Moodle-Prüfung: Zeit 90’ (Zeitdruck)
  + ERD
  + Select mit Where-Clauses und Function-Calls
  + Date und Timestamp Formatierung
  + Inner und Outer Joins
  + Eigene Views und SQL-Statements nach vorgegebenen Requirements erstellen

Nachbereitung

* Aufgaben lösen (Views)

|  |  |
| --- | --- |
| **Thema Block 6: DDL: Create Schema, Tables and Relations** | |
| Datum | Di 23.5.2023 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* DDL Befehle (CRUD Operationen auf Meta-Data)

Inhalte (Folien 45 – 48)

* CREATE
* ALTER
* DELETE
* Constraints

Nachbereitung

* Neues Schema für Adressverwaltung designen und mit SQL definieren
* Tabellen mit Test-Daten füllen (Scripts erstellen)
* Normalisieren und Daten migrieren

|  |  |
| --- | --- |
| **Thema Block 7: DB access via Python / User-Rights** | |
| Datum | Di 13.6.2023 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Was sind die Elemente in einem Connection-String?
* Wie werden Zugriffsrechte in MySQL definiert

Inhalte (Folien 49 – 51)

* 2 Applikations-User für eigenes Schema (Manager: read/write auf alles, Normal: Access nur auf Views und Procedures) definieren.
* Python IDE vorbereiten, dass auf DB verbunden werden kann und Connection erstellen.
* Folgende DML Befehle aus Python ausführen lassen:
  + select, insert, function auf eine Tabelle und eine View
  + call Storded-Procedures

|  |  |
| --- | --- |
| **Thema Block 8: 2. Leistungsnachweis** | |
| Datum | Di 27.6.2023 |
| Zeit | **17:30 – 20:45** |

Inhalte

* Moodle-Prüfung: Zeit 3h
  + DDL und DML Befehle in allen Varianten
  + Functions und Stored-Procedures
  + Transactions
  + Python Access DML und DDL, sowie Stored-Procedures calls