

## Lehrveranstaltung "Erweiterte Datenbanktechnologien / Medienarchive" Aufgabenstellungen für die Übungen

### Aufbau einer Datenbasis im DBMS SYBASE ASE

In der Tabelle **Mitarbeiter** werden die Daten für jeden Mitarbeiter gespeichert. In der Tabelle **Projekt** sind die Daten für jedes Projekt abgelegt. Die Leiternr muss einer Mitnr aus der Tabelle **Mitarbeiter** entsprechen. In der Tabelle **MiPro** wird festgehalten welcher Mitarbeiter an welchem Projekt arbeitet. Die Spalte Istvzae enthält hierbei den tatsächlichen Anteil der Arbeitszeit des Mitarbeiters, Planvzae den geplanten. Ist ein Projekt beendet, wird der Datensatz nicht aus der Tabelle Projekt gelöscht. Aber nachdem sichergestellt wurde, dass keine Mitarbeiter mehr im Projekt arbeiten, wird die Leiternr für das Projekt gelöscht.

#### **Mitarbeiter**

Spaltenname	Typ	Bemerkung
Mitnr	char(5)	Primärschlüssel, Not Null
Name	char(10)	
Vorname	char(10)	
Ort	char(20)	
Gebdat	date	
Beruf	char(15)	
Telnr	char(12)	

#### **Projekt**

Spaltenname	Typ	Bemerkung
Pronr	char(2)	Primärschlüssel, Not Null
Proname	char(15)	
Beschreibung	char(30)	
Leiternr	char(5)	Null

#### **MiPro**

Spaltenname	Typ	Bemerkung
Mitnr	char(5)	Primärschlüssel, Not Null
Pronr	char(2)	
Istvzae	float	
Planvzae	float	

### ***Hinweis zu SQL-Befehlen:***

*Kopieren* einer Tabelle (Datendefinition und Daten). Es ist zu beachten, dass Indizes und Constraints (incl. Primär- und Fremdschlüssel) nicht mit kopiert werden:

```
select * into <Neue Tabelle> from <Quelle>
```

*Importieren* der Daten in eine Tabelle aus einer anderen Tabelle:

```
insert into <Ziel> select * from <Quelle>
```

In beiden Fällen kann das \* im Select der Anweisung natürlich durch eine Spaltenliste ersetzt werden.

### **1. Aufgabenkomplex: Aufbau der Datenbank**

1. Legen Sie die Tabelle **Mitarbeiter** an und importieren Sie die Daten aus der Tabelle **depotN.dbo.quelleDB3Mitarbeiter**.
2. Kopieren Sie die Tabelle **depotN.dbo.quelleDB3Projekt** in die Tabelle **Projekt**. Legen Sie den erforderlichen Primärschlüssel an.
3. Kopieren Sie die Tabelle **depotN.dbo.quelleDB3MiPro** in die Tabelle **MiPro**. Legen Sie den erforderlichen Primärschlüssel an.
4. Skizzieren Sie die Beziehungen zwischen den Tabellen auf der ersten Seite und legen Sie die entsprechenden Fremdschlüssel an.
5. SQL-Abfragen:
  - a. Geben Sie alle Mitarbeiter zum Projekt „31“ aus. (Ausgabe: Mitnr, Name, Vorname, Projektname)
  - b. Geben Sie alle Mitarbeiter aus, deren Gesamtsumme der Istvzae über 0,8 liegt (Ausgabe: Mitnr, Name, Summe der Istvzae)
6. Schreiben Sie eine Prozedur. Diese Prozedur soll ohne Cursor realisiert werden. Alle Ausgaben von Daten sollen über ein Select erfolgen. Für eine, als Parameter übergebene Pronr sollen alle Informationen (Name, Vorname, Telefon) zum Projektleiter ausgegeben werden. Des Weiteren sollen alle Mitarbeiter des Projektes (Mitnr, Name, Vorname) angezeigt werden. Sollte die Pronr ungültig sein, so soll eine Fehlermeldung erscheinen.
7. Schreiben Sie einen Trigger, der sicherstellt, dass die Leiternr nicht aus der Tabelle **Projekt** gelöscht werden kann, solange in diesem Projekt noch Mitarbeiter beschäftigt sind.

## **Lehrveranstaltung "Erweiterte Datenbanktechnologien / Medienarchive" Aufgabenstellungen für die Übungen**

### **Call-Schnittstellen von DBMS**

#### ***Hinweise zu den Aufgaben 2,3,4 und 6:***

Um das Programm zu compilieren und auszuführen, loggen Sie sich auf der ilux150 ein.

Zur Behandlung von Nachrichten und Fehlermeldungen werden die Funktionen `error_handler` und `message_handler` benötigt. Fügen Sie diese Funktionen aus der Datei `/u/sw_sybase/DBST_WS1112/dateien_prakt/handler_funktionen.txt` in Ihr c-File ein.

Kopieren Sie die Dateien `makefile`, `sybdbex.h` und `sybopts.sh` aus dem Verzeichnis `/u/sw_sybase/DBST_WS1112/dateien_prakt` in Ihr Projektverzeichnis und passen Sie das `makefile` entsprechend an.

Zum Compilieren und Ausführen des Programms müssen vorher die Umgebungsvariablen mit

**`. /u/sw_sybase/env_syb/linux_sybase155`**

gesetzt werden.

Informationen zu Open-Server/Open-Client - DB-LibraryTM/C Reference Manual::

<http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc32600.1500/pdf/dblib.pdf>

#### **2. Aufgabenkomplex: CLI - Prinzipielle Programmstruktur**

Es ist ein C-Programm zu schreiben, das mit Mitteln der DB-Library C den Zugriff auf die Tabellen `Mitarbeiter`, `MiPro` und `Projekt` realisiert. Die nachfolgenden Teilaufgaben sollen als Unterprogramme eines Rahmenprogramms geschrieben werden und so schrittweise die Leistung des Programms erweitern.

1. Anzeige aller verschiedenen Berufe für Auswahlzwecke.  
Auswahl eines Berufes.
2. Anzeige der Mitarbeiternummern, Namen und Vornamen für den ausgewählten Beruf.  
Auswahl einer Mitarbeiternummer.
3. Anzeige folgender Daten für die ausgewählte Mitarbeiternummer:  
Name, Vorname, Ort, Beruf, Geburtstag, Telefonnummer, Projektname, Istvzae und Planvzae.

Leereingabe soll den Übergang in die übergeordnete Ebene bewirken.

### 3. Aufgabenkomplex: CLI - Benutzung von Stored Procedures

Schreiben Sie ein C-Programm, das mit Mitteln von DB-Library-C:

1. die in der Tabelle **Mitarbeiter** vorhandenen unterschiedlichen Orte anzeigt,
2. vom Nutzer die Eingabe eines Ortes verlangt,
3. für den ausgewählten Ort das Durchschnittsalter der dort wohnenden Mitarbeiter ermittelt und anzeigt , indem es eine stored procedure nutzt.

Schreiben Sie die **stored procedure**, die für einen übergebenen Ort das Durchschnittsalter der im Ort wohnenden Mitarbeiter zurückgibt.

### 4. Aufgabenkomplex: CLI - Lesen/Schreiben von CLOBs

Vorbereitung:

Ergänzen Sie die Tabelle **Mitarbeiter** um eine Spalte **Lebenslauf** vom Datentyp **text**  
( mit SQL-Befehlen unter JISQL).

Aufgaben

Schreiben Sie ein CD-Library C-Programm mit folgenden Leistungen:

1. Eingabe einiger Zeilen Beispielttext mit Hilfe von **dbwritetext**.  
Vergessen Sie nicht, vorher einen Textpointer z.B. mit **update . . .** unter JISQL zu initialisieren.
2. Kontrollanzeige des Beispielttextes mit Hilfe von **dbreadtext** und C-Funktionen nach Bildschirmanforderung.  
Bitte beachten Sie, daß **dbreadtext** in der Benutzung dem **dbnextrow** entspricht, es wird für die Datentypen **text** und **image** angewandt.

## Lehrveranstaltung "Erweiterte Datenbanktechnologien / Medienarchive" Aufgabenstellungen für die Übungen

### Verteilte Datenbanksysteme

#### 5. Aufgabenkomplex: : Verteilte Datenbanken - CIS Nutzung

Mit Hilfe des Sybase-Servers *syb150* sollen Tabellen, die auf den Servern *syb150* und *syb125c* liegen, transparent ausgewertet werden.

Dazu sind folgende Vorarbeiten zu leisten:

- Sichern Sie zuvor ihre Tabellen, sodass problemlos zu einer früheren Aufgabenstellung/Lösung zurückgekehrt werden kann: **sp\_rename Mitarbeiter, Mitarbeiter\_old**  
Zur Gewährleistung eines einheitlichen Ausgangszustands ist die Tabelle **Mitarbeiter** aus der Datenbank **depot** (*syb150*) in ihre Beispieldatenbank **[ii|im|iwi]s9999** zu kopieren:  
Befehl zum Kopieren: **select \* into Mit from depot.dbo.Mit**
- Einrichten einer neuen Tabelle auf der **syb125c** in der jedem Mitarbeiter eine Gehaltsgruppe zugewiesen wird.  
**Mitgeh (Mitnr char(5), Gehgr char(3))**  
Einrichten einer neuen Tabelle auf der **syb125c** in der Gehälter und die Gehaltsgruppen enthalten sind.  
**Gehaelter(Gehalt float, Gehgr char(3))**
- "Bekanntmachen" der **syb150** mit den entfernt liegenden Tabellen ( "Definieren der Verbindung" ) --> table mapping -->  
Gehälter und  
Mitarbeitergehaltsgruppen auf **syb125c . [ii|im|iwi][Jahrgang]s99999 . dbo . Mitgeh** und **syb125c . [ii|im|iwi][Jahrgang]s99999 . dbo . Gehaelter** ( **sp\_addtabledef ...** ).
- "Bekanntmachen" des Servers *syb150* mit den Definitionen (Metadaten) der entfernt liegenden Tabellen -->data mapping -->  
Gehälter und Mitarbeitergehaltszuordnung (**create existing table ...** )

Informieren Sie sich mit Hilfe geeigneter Systemprozeduren (**sp\_help...**) über die vorhandenen Server und über Aufbau und Lage der eingerichteten Tabellen.

**Nach diesen Vorarbeiten sollen folgende Auswertungen vorgenommen werden:**

1. Ermittlung des Gehaltes für den Herrn Fuchs ( Anzeige Name, Ort, Gehalt)
2. Ermittlung des Durchschnittsgehaltes der in Dresden wohnenden Mitarbeiter ( Anzeige Ort, Durchschnittsgehalt )

3. Ermittlung des Mitarbeiters mit dem höchsten Gehalt (Anzeige Name, Gehalt)
4. Ermittlung des Mitarbeiters mit dem niedrigsten Gehalt (Anzeige Name, Gehalt)
5. Geben Sie Frau Elster eine Gehaltserhöhung ( zahlenmäßig nächstniedrigere Gehaltsstufe! )  
Formulieren Sie dabei nur ein SQL-Statement

**Zum Sichern der Datenintegrität legen Sie bitte für die folgenden 2 Aufgaben je einen Trigger an! Beachten Sie bei der Erstellung von Triggern folgende Restriktionen:**

6. Einfügen des Mitarbeiters Meyer mit der Gehaltsgruppe 1a und der Mitarbeiternummer 120  
in den Diskursbereich (restliche Daten nach Ihrer Wahl).
7. Löschen Sie alle Daten des Mitarbeiters Rabe.

### **6. Aufgabenkomplex: Two-Phase-Commit (programmierter Fall)**

Auf **syb150** und **syb125c** existiert je eine **Datenbank** [ii|im|iww][Jahrgang]s99999 mit einer Tabelle **Mitarbeiter**.  
(s99999 entspricht Ihrer Bibliotheksnummer, Beispiel für DB-Namen: iw09s12345).

Schreiben Sie ein C-Programm unter Benutzung der DB-Library, das in den inhaltlich gleichen Tabellen **Mitarbeiter** , die von zwei verschiedenen Servern verwaltet werden, das Alter des Mitarbeiters Fuchs in beiden Mit in gleicher Weise während einer Transaktion ändert.

**Sichern Sie die Integrität der Daten mit Hilfe des Two-Phase-Commit.**

Überprüfen sie in der Tabelle **syb150.spt\_committab** den Status ihrer Transaktion auf beiden Servern: **select commid, start, totnum, outnum, status, applname, xactname from spt\_committab [oder by start]**

**Vorbereitung:**

Kopieren Sie die Tabelle Mitarbeiter aus der Datenbank depot in Ihre Datenbanken [ii|im|iww][Jahrgang]s99999. Die Datenbank depot wird auf der syb150 und der syb125c bereitgestellt.

Sichern Sie zuvor ihre Tabellen, sodass problemlos zu einer früheren Aufgabenstellung/Lösung zurückgekehrt werden kann: **sp\_rename Mitarbeiter, Mitarbeiter\_old**

Befehl zum Kopieren: **select \* into Mitarbeiter from depot.dbo.Mitarbeiter**

### **8. Aufgabenkomplex: Replikation (folgt im Komplex MS SQL-Server))**