



EMPLOYEE DB ADVANCED



AUSGANGSLAGE

Deine Firma führt Archivdatenbanken der Mitarbeiter. Aus irgendeinem Grund gibt es keine Archivdatenbank für das Jahr 1995. Der Chef wünscht, dass der Schaden so schnell wie möglich behoben wird und eine Archivdatenbank aus den aktuellen Daten abgeleitet wird. (Ausgangsdatenbank: Employee-Datenbank)

Zum Glück hat sich an der Firmenstruktur, den Standorten und Abteilungen nichts verändert. Nach gemeinsamen Überlegungen mit den anderen Mitarbeitern kommt ihr zum Entschluss folgende Schritte durchführen zu müssen:

- Eine neue Datenbank erstellen
- Die Tabellen der aktuellen Datenbank übertragen
- Die Inhalte, die gleich sind, 1:1 übertragen (locations, countries, regions, departments, jobs)
- Die Mitarbeiter, deren Einstellungsdatum vor 1996 liegt übertragen
- Überprüfen, ob es Manager gibt, die zu dieser Zeit noch nicht in der Firma gearbeitet haben, wenn das der Fall ist, den Manager auf 100 (Steven King) setzen
- Die Einträge der Jobhistory deren Enddatum nach 31.12.1995 liegt und das Startdatum vor 1996 liegt als aktuellen Job und aktuelle Abteilung in der Employees-Tabelle eintragen



DATENBANK ERSTELLEN UND INHALTE ANPASSEN

Löse alle Aufgaben ausschließlich mit SQL Statements und speichere diese in deiner Console. Prüfe deine Statements bevor du die Datenbank wirklich veränderst (Tx:Manual). Formuliere zuerst ein SELECT-Statement und entwickle erst dann ein UPDATE-Statement daraus. Verwende Nested Querys!



WEBZUGANG

Nun will der Chef auch noch eine kleine Webseite auf die sich die Mitarbeiter aus dem Jahr 1995 einloggen können und sehen können, welchen Job sie hatten und wie die Abteilung hieß. Füge zur Employee-Tabelle ein Passwort hinzu. Das Passwort darf keine Null-Werte enthalten und das soll nun zu Beginn gleich dem Nachnamen des Mitarbeiters gesetzt werden.

Erstelle eine kleine Useroberfläche womit sich der Mitarbeiter einloggen kann und sieht, welchen Job er im Jahr 1995 hatte und den Namen der Abteilung, in der er gearbeitet hat. Achte dabei, dass der Login sicher vor **SQL INJECTION** ist.