Projekt Zoo II

Jan Siekierka, Sandro Zeneljaj, Kinh Bac Nguyen



**Berufliches Schulzentrum für Elektrotechnik Dresden**

Strehlener Pl. 2, 01219 Dresden

Inhaltsverzeichnis

1. Analyse der vorhandenen Datenbasis 3
2. Backupmechanismus 3
3. Vergleich Planung mit vorhandener Datenbasis 3
4. Datenbankdiagramm (relationales Modell) 4
5. Planung Views (Abfragen) 4
6. Berichterstellung 6
7. Seriendrucke 6
8. Rechtevergabe 7
9. Datenbankanwendung 8
10. Anlagen 9
11. Quellen 13
12. Selbstständigkeitserklärung 14
13. Analyse der vorhandenen Datenbasis

**Tabelle(Primärschlüssel, Fremdschlüssel, Attribut)**

Abteilung(Abt\_ID, Bezeichnung)

Art(Tierart\_ID, Tierart, Klasse)

Futter(Futter\_ID, Futter\_Name, Futter\_Beschreibung)

Futtermenge(Futter\_Nr, Rations\_Nr, Menge)

Futterration(Rations\_ID, Tiernummer, Häufigkeit)

Gehege(Gehege\_ID, Bezeichnung, Objekt\_Nr)

Gehegebetreuung(Pfleger\_Nr, Gehege\_Nr)

Klasse(Klasse\_ID, Klassenname)

Mitarbeiter(MitArb\_ID, Abt\_NR, Name, Vorname, Strasse, PLZ, Ort, Taetigkeit)

Objekte(Objekt\_ID, Bezeichnung, Objekt\_Art)

Tiere(TierNr, Name, Bezeichnung, Art, Anzahl, Gehege\_Nr)

1. Backupmechanismus  
     
   Für das tägliche Backup wird ein Bordmittel von Microsoft SQL Server Management Studio verwendet. Auf einem Backupvolume wird die Datenbank als SQL Skript gespeichert und täglich aktualisiert.   
   Dazu werden folgende Schritte durchgeführt:
   1. Objekt-Explorer > Datenbanken > Zoo > Rechtsklick > Tasks > Skripts generieren
   2. Registerkarte Einführung > Weiter
   3. Registerkarte Objekte auswählen > [x] Skripterstellung für gesamte Datenbank und alle Datenbankobjekte auswählen  
      Weiter
   4. Registerkarte Skripterstellungsoptionen festlegen   
      > [x] Als Skriptdatei speichern  
      > Zu generierende Dateien: [x] Einzelne Skriptdatei  
      > Dateiname: *Speicherpfad der Datei auf dem Backupstorage einfügen*   
      > Speichern als: [x] Unicode-Text
   5. Registerkarte Zusammenfassung > Weiter
   6. Skripts speichern > Kurz warten, bis Dateierstellung erfolgreich abgeschlossen ist   
      > Fertig Stellen
2. Vergleich Planung mit vorhandener Datenbasis

Folgende Änderungen wurden vorgenommen:

* Art → Tierart
  + Tierart → Bezeichnung
  + Klasse → Tiergattung\_Nr
* Futterration
  + Tiernummer → Tier\_Nr
* Gehege
  + + letzte\_Reinigung
* Gehegebetreuung
  + Pflege\_Nr → Mitarbeiter\_Nr
* Klasse → Tierklasse
  + Klasse\_ID → Tierklasse\_ID
  + Klassenname → Bezeichnung
* Mitarbeiter
  + MitArb\_ID → Mitarbeiter\_ID
  + - Abt\_NR
  + Taetigkeit → Tätigkeit\_Nr
  + + Geschlecht
  + + aktiv
* Objekte
  + Objekt\_Art → Objektart\_Nr
* Tiere
  + TierNr → Tier\_ID
  + Bezeichnung → Tierart\_Nr
  + - Art
* + Futtertransport
* + Lager
* + Lagerart
* + MedizinischeUntersuchung
* + Objektart
* + Tätigkeit
* + Tiergattung
* + Tiertransport
* + Einkauf

1. Datenbankdiagramm (relationales Modell)

siehe Anlage 1

1. Planung Views (Abfragen)

Für die Views wurden folgende Abfragen geplant. Die Views, welche in der DB realisiert werden, sind mit einem x markiert.

* 1. handwerk.Gehege\_Objekt\_Anzahl (x)

SELECT o.Bezeichnung, COUNT(g.Gehege\_ID) AS Anzahl

FROM Objekte o, Gehege g

WHERE o.Objekt\_ID = g.Objekt\_Nr

GROUP BY o.Bezeichnung;

* 1. tierpflege.Futterpläne (x)

SELECT t.Tier\_ID, t.Name, ta.Bezeichnung, f.Futter\_Name, fm.Menge, fr.Häufigkeit

FROM Futter f, Futtermenge fm, Futterration fr, Tiere t, Tierart ta

WHERE f.Futter\_ID=fm.Futter\_Nr

AND fr.Rations\_ID=fm.Rations\_Nr

AND t.Tier\_ID=fr.Tier\_Nr

AND ta.Tierart\_ID=t.Tier\_ID;

* 1. Futtertransport
  2. verwaltung.Futtertyp\_Einkauf (x)

SELECT f.Futter\_Name, SUM(e.Menge) AS Menge

FROM Futter f, Einkauf e

WHERE f.Futter\_ID=e.Futter\_Nr

GROUP BY f.Futter\_Name;

* 1. Futtervergabe
  2. Gehege\_Futterration\_Menge
  3. tierpflege.Gehegebetreuung\_Mitarbeiteranzahl (x)

SELECT g.Bezeichnung, COUNT(m.MitArb\_ID) AS Anzahl

FROM Gehege g, Gehegebetreuung gb, Mitarbeiter m

WHERE g.Gehege\_ID=gb.Gehege\_Nr

AND m.MitArb\_ID=gb.Mitarbeiter\_Nr

GROUP BY g.Bezeichnung;

* 1. tierpflege.Gehege\_Tieranzahl (x)

SELECT g.Bezeichnung, COUNT(t.Tier\_ID) AS Anzahl

FROM Gehege g, Tiere t

WHERE g.Gehege\_ID=t.Gehege\_Nr

GROUP BY g.Bezeichnung;

* 1. Tierart\_Tierbefunde
  2. Tiertransport
  3. Tierklasse\_Gehegeanzahl
  4. Tierarten\_Gehegeanzahl
  5. Tiergattungen\_Gehegeanzahl
  6. tierpflege.Tierart\_MedizinischeUntersuchungen (x)

SELECT ta.Bezeichnung, COUNT(m.MedUntersuchung\_ID) AS Anzahl

FROM tierpflege.MedizinischeUntersuchung m, Tierart ta, Tiere t

WHERE t.Tierart\_Nr = ta.Tierart\_ID

GROUP BY ta.Bezeichnung;

* 1. tierpflege.Tierklasse\_Tieranzahl (x)

SELECT tk.Bezeichnung, COUNT(t.Tier\_ID) as Anzahl

FROM Tiere t, Tierart ta, Tiergattung tg, Tierklasse tk

WHERE tg.Tierklasse\_Nr=tk.Tierklasse\_ID

AND ta.Tiergattung\_Nr=tg.Tiergattung\_ID

AND t.Tierart\_Nr=ta.Tierart\_ID

GROUP BY tk.Bezeichnung;

* 1. Tiergattung\_Tieranzahl
  2. tierpflege.Tierart\_Tieranzahl (x)

SELECT ta.Bezeichnung, COUNT(t.Tier\_ID) AS Anzahl\_Tiere

FROM Tiere t, Tierarten ta

WHERE ta.Bezeichnung=t.Bezeichnung

GROUP BY ta.Bezeichnung;

* 1. hr.Abteilung\_Mitarbeiteranzahl (x)

SELECT a.Bezeichnung, COUNT(m.MitArb\_ID) AS Anzahl

FROM Mitarbeiter m, Abteilung a

WHERE m.Abt\_NR = a.Abt\_ID

GROUP BY a.Bezeichnung;

* 1. Abteilung\_Tätigkeit
  2. hr.Tätigkeit\_Mitarbeiteranzahl (x)

SELECT t.Bezeichnung, COUNT(m.MitArb\_ID) AS Anzahl

FROM Mitarbeiter m, Taetigkeit t

WHERE m.Taetigkeit\_Nr=t.Taetigkeit\_ID

GROUP BY t.Bezeichnung;

* 1. hr.Wohnort\_Mitarbeiteranzahl (x)

SELECT m.Ort, COUNT(m.MitArb\_ID) as Anzahl

FROM Mitarbeiter m, Mitarbeiter ma

WHERE m.Ort=ma.Ort

AND m.MitArb\_ID=ma.MitArb\_ID

GROUP BY m.Ort;

* 1. Lager\_Futtertypanzahl
  2. Lager\_Futtermenge
  3. Lagerbestand\_Gesamt
  4. handwerk.Objektart\_Anzahl (x)

SELECT oa.Bezeichnung, COUNT(o.Objekt\_ID) AS Anzahl

FROM Objekte o, Objektart oa

WHERE o.Objektart\_Nr = oa.Objektart\_ID

GROUP BY oa.Bezeichnung;

1. Berichterstellung

Die Berichte wurden nach der Anleitung aus dem Unterrichtsskipt „05\_07 Erstellen von Berichten\_BI\_Visual\_Studio.pdf“ erstellt.

* 1. Gehege Objekt Anzahl
     1. Bezeichnung
     2. Anzahl
  2. Tätigkeiten Mitarbeiteranzahl
     1. Bezeichnung
     2. Anzahl
  3. Wohnort Mitarbeiteranzahl
     1. Ort
     2. Anzahl
  4. Gehegebetreuung Mitarbeiteranzahl
     1. Bezeichnung
     2. Anzahl
  5. Futterpläne
     1. Name
     2. Bezeichnung
     3. Futter Name
     4. Menge
     5. Häufigkeit

1. Seriendruck  
   Zwei Seriendruckentwürfe wurden in dem Projekt realisiert  
   - Einladung zur Weihnachtsfeier (seriendruck\_weihnachtsfeier.docx)  
   - Neuigkeiten zur Saisoneröffnung (seriendruck\_neues\_zur\_saisoneröffnung.docx)
2. Rechtevergabe

Schemata (schemas) und dazugehörige Datenbankobjekte

**verwaltung**

* View.Futtertyp\_Einkauf
* Einkauf

**hr**

* View.Abteilung\_Mitarbeiteranzahl
* View.Tätigkeit\_Mitarbeiteranzahl
* View.Wohnort\_Mitarbeiteranzahl

**tierpflege**

* View.Gehege\_Tieranzahl
* View.Tierart\_Tieranzahl
* View.Tierklasse\_Tieranzahl
* View.Futterpläne
* View.Gehebetreuung\_Mitarbeiteranzahl

**handwerk**

* View.Objektart\_Anzahl
* View.Gehege\_Objekt\_Anzahl

Datenbankrollen (database role)

|  |  |
| --- | --- |
| **Name** | **Rechte** |
| personalerRole | SELECT für Schema hr;  SELECT, UPDATE, INSERT für Tabelle dbo.Mitarbeiter |
| pflegerRole | SELECT für Schema tierpflege; |
| arztRole | SELECT, UPDATE, DELETE für Tabelle dbo.MedizinischeUntersuchung |
| handwerkerRole | SELECT für Schema handwerk |
| verwalterRole | SELECT für Schema verwaltung |

Anmeldungen/Logins + Benutzer

|  |  |  |  |
| --- | --- | --- | --- |
| **Anmeldename** | **Benutzername** | **PW für Anmeldung** | **Zugewiesene Rollen** |
| personalabteilung | personalabteilungUser | abc123 | personalerRole |
| pfleger1 | pflegerUser | abc123 | pflegerRole |
| tierarzt | tierarztUser | arzt123 | arztRole, pflegerRole |
| handwerker | handwerkerUser | handwerker123 | handwerkerRole |
| verwaltung | verwaltungUser | verwaltung123 | verwalterRole |

Mithilfe von diversen SQL-Befehlen werden die Zugriffsrechte umgesetzt (siehe Anlage 2).

1. Datenbankanwendung (CRUD-App)

Es wurde eine Anwendung für den Nutzer „personalabteilungUser“ umgesetzt.

Er stellt einen Mitarbeiter aus der Personalabteilung dar, der alle in der Datenbank gespeicherten Mitarbeiterdaten sehen, aktualisieren und neue hinzufügen darf. Was er nicht darf, ist einen Eintrag zu löschen. Hier darf er Mitarbeiter nur auf aktiv bzw. inaktiv setzen. Daraus ergeben sich auch die Navigationspunkte (siehe Anhang 3).

Für den Anwender ist also die graphische Oberfläche (GUI) in Windows Forms zu sehen (Frontend), während Datenbankabfragen, genau genommen Datenbankoperationen auf die Datenbank-Tabelle „dbo.Mitarbeiter“, über die SqlConnection-Klasse aus der .NET-Bibliothek „SqlClient“ realisiert wird.

10. Anlagen

Anlage 1 – relationales Datenbankdiagramm

Ein Bild, das Text, Diagramm, Plan, technische Zeichnung enthält.

Automatisch generierte Beschreibung

Anlage 2 – SQL-Skripte für die Rechtevergabe

-- Erzeugung Logins mit Passwort

CREATE LOGIN personalabteilung WITH PASSWORD='abc123', CHECK\_POLICY = OFF;

CREATE LOGIN pfleger WITH PASSWORD='abc123', CHECK\_POLICY = OFF;

CREATE LOGIN tierarzt WITH PASSWORD='arzt123', CHECK\_POLICY = OFF;

CREATE LOGIN handwerker WITH PASSWORD='abc123', CHECK\_POLICY = OFF;

CREATE LOGIN verwaltung WITH PASSWORD='abc123', CHECK\_POLICY = OFF;

-- Erzeugung Benutzer mit Verknüpfung zum Login und Schema

CREATE USER personalabteilungUser FOR LOGIN personalabteilung WITH DEFAULT\_SCHEMA = dbo;

CREATE USER pflegerUser FOR LOGIN pfleger WITH DEFAULT\_SCHEMA = dbo;

CREATE USER tierarztUser FOR LOGIN tierarzt WITH DEFAULT\_SCHEMA = dbo;

CREATE USER handwerkerUser FOR LOGIN handwerker WITH DEFAULT\_SCHEMA = dbo;

CREATE USER verwaltungUser FOR LOGIN verwaltung WITH DEFAULT\_SCHEMA = dbo;

-- Erzeugung Benutzerrollen

CREATE ROLE personalerRole;

CREATE ROLE pflegerRole;

CREATE ROLE arztRole;

CREATE ROLE handwerkerRole;

CREATE ROLE verwalterRole;

-- Zuweisung Mitgliedschaft von Benutzern zu Benutzerrollen  
-- Definieren von Zugriffsrechten für Benutzerrollen

ALTER ROLE personalerRole ADD MEMBER personalabteilungUser;

GRANT SELECT ON SCHEMA :: hr TO personalerRole;

GRANT SELECT, UPDATE, INSERT ON dbo.Mitarbeiter TO personalerRole;

ALTER ROLE pflegerRole ADD MEMBER pflegerUser;

GRANT SELECT ON SCHEMA :: tierpflege TO pflegerRole;

GRANT EXECUTE ON SCHEMA :: tierpflege TO pflegerRole;

ALTER ROLE arztRole ADD MEMBER tierarztUser;

GRANT SELECT, UPDATE, DELETE, INSERT ON dbo.MedizinischeUntersuchung TO arztRole;

ALTER ROLE handwerkerRole ADD MEMBER handwerkerUser;

GRANT SELECT ON SCHEMA :: verwaltung TO verwalterRole;

Anlage 3 – Navigation in Anwendung für einen Mitarbeiter der Personalabteilung

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

11. Quellen

- Projektabgabe „Zoo I“

- Unterrichtsmaterial

- (17.02.2024). CREATE LOGIN (Transact-SQL. https://learn.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql?view=sql-server-ver16

12. Selbstständigkeitserklärung   
  
  
Hiermit erklären wir, dass wir die vorliegende Arbeit selbstständig und ohne unerlaubte fremde Hilfe angefertigt, keine anderen als die   
angegebenen Quellen und Hilfsmittel verwendet und die den verwendeten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht haben.   
  
  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Ort, Datum  
  
  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Unterschriften