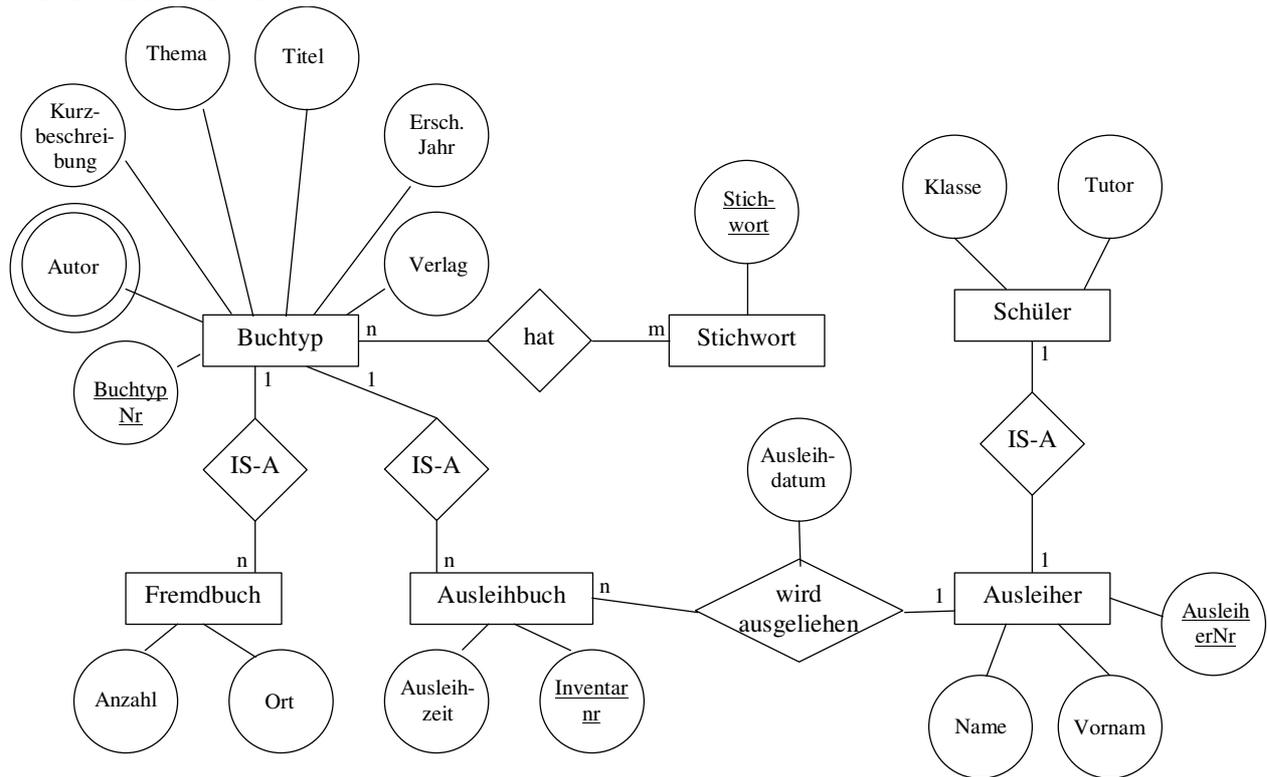


ARBEITSBLATT ZUR UMSETZUNG DES REL. MODELLS IN ACCESS

Sie kennen noch das Entity-Relationship-Diagramm unserer Bücherei-Verwaltung aus einem der ersten Arbeitsblätter.



Die Umsetzung im relationalen Modell hatte folgendes Aussehen:

Buchtyp (Buchtyp-Nr, Titel, Erscheinungsjahr, Verlag, Thema, Kurzbeschreibung)

Autor (Buchtyp-Nr, Autor)

Ausleihbuch (InventarNr, Buchtyp-Nr, Ausleihzeit)

Stichwort (StichwortNr, Stichwort)

Ausleiher (Ausleiher-Nr, Name, Vorname)

Gehörtzu (Buchtyp-Nr, StichwortNr)

Fremdbuch (Buchtyp-Nr, Ort, Anzahl)

Schüler (Ausleiher-Nr, Klasse, Tutor)

Ausleihe (Ausleiher-Nr, InventarNr, Ausleihdatum)

Zur Erinnerung:

Das Mehrfachattribut *Autor* wurde in einen eigenen Entity-Typen umgewandelt.

Die n:m-Beziehung zwischen *Buchtyp* und *Stichwort* muss durch eine Beziehungs-Relation *Gehörtzu* in zwei obligatorische 1:n-Beziehungen aufgelöst werden. Um die Stichwörter nicht in zwei Tabelle aufzuführen wurde ein Schlüssel *StichwortNr* eingeführt.

Die 1:n-Beziehung zwischen *Ausleihbuch* und *Ausleiher* ist nicht obligatorisch (ein Ausleihbuch muss nicht zwangsläufig ausgeliehen sein). Deshalb muss die Beziehung in einer eigenen Beziehungsrelation *Ausleihe* festgehalten werden.

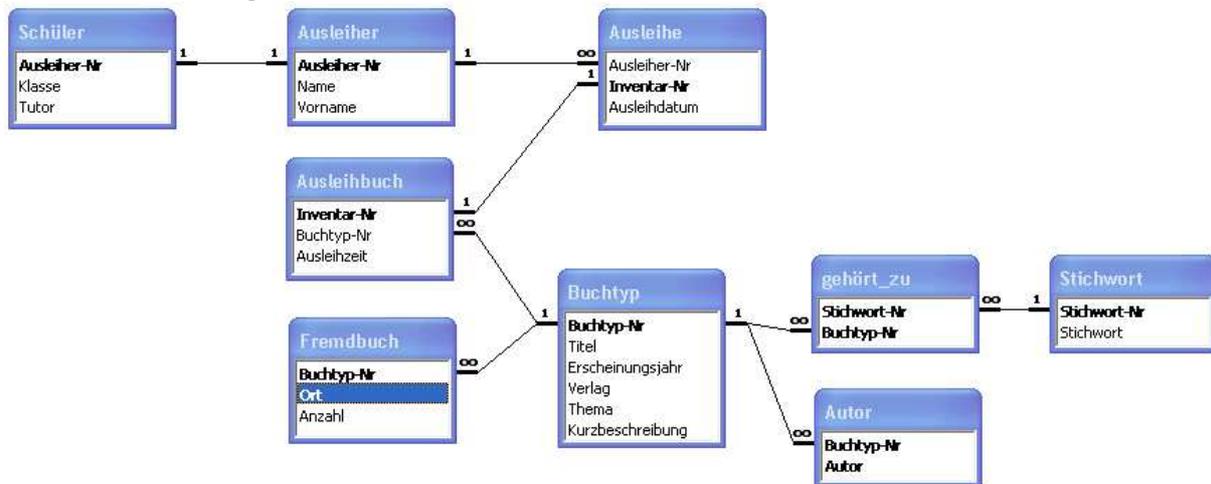
Im folgenden Wollen wir nun dieses Modell in Access umsetzen.

Aufgabe 1: Erstellen Sie zu allen Entity-Typen eine Tabelle in Access, so wie rechts abgebildet. Wählen Sie in den jeweiligen Tabellen für die Attribute sinnvolle Namen und Datentypen.



Aufgabe 2: Importieren Sie für jede Tabelle die Datensätze aus den entsprechenden Excel-Tabellen. Sie finden die Tabellen auf dem Tausch-Server.

Aufgabe 3: Setzen Sie die Tabellen zueinander in Beziehung. Denken Sie an referentielle Integrität:



Aufgabe 4: Erstellen Sie Abfragen mit folgenden Zielen:

- a) Ausgabe aller Bücher (Thema, Titel und Verlag), die dem Thema Programmieren zuzuordnen sind, welche entweder nach 1995 oder im Verlag "Microsoft" erschienen sind.

Hinweis: Thema Programmieren kann bedeuten: "Programmieren PASCAL". Deshalb soll das Thema lediglich mit Programmieren anfangen:

Kriterien: `Wie "Programmieren" & "*"`

- b) Ausgabe aller Bücher (Titel, Autor und Erscheinungsjahr), welche zwischen 1985 und 1992 erschienen sind und von einem frei wählbaren Autor geschrieben wurden.

Hinweis: Einen frei wählbaren Autor erhält man durch das Kriterium:

Kriterien: `Wie [Autor eingeben] & "*" ..`. Der Benutzer erhält ein Fenster mit der Aufforderung "Autor eingeben".

- c) Ausgabe aller Bücher (Titel, Autor), welche einem frei wählbaren Stichwort zuzuordnen sind.

Hinweis: Der Benutzer soll auch nur Teile eines Stichworts eingeben können.

- d) Ausgabe aller Bücher (Autor, Titel und Ausleihdatum) eines frei wählbaren Schülers

- e) Ausgabe aller Schüler (Name, Vorname), die ihre Ausleihfrist überschritten haben. (Wir schreiben heute den 30.11.2006)

- f) Ausgabe der Bestands-Anzahlen eines frei wählbaren Buchtyps, sowohl insgesamt als auch ausgeliehen.

Hinweis: Die Verknüpfung zwischen *Buchtyp* und *Ausleihbuch* muss nun so gestaltet werden, dass alle Einträge der Tabelle *Buchtyp* angezeigt werden, auch die, welche nicht ausgeliehen sind. Dies erreicht man, indem man die Verknüpfungseigenschaft von INNER JOIN auf LEFT JOIN umstellt (SQL-Ansicht der Abfrage).

- g) Erstellung einer neuen Tabelle, die alle abgeschlossenen Ausleihvorgänge umfasst, die vor dem 15.03.2005 getätigt wurden.

Hinweis: Stellen Sie den Abfragetyp um auf eine Tabellenerstellungsabfrage.

- h) Löschung aller Datensätze aus der eben neu erstellten Tabelle.

Hinweis: Stellen Sie den Abfragetyp um auf eine Löschartabfrage.

Aufgabe 5: Schauen Sie sich alle Abfragen in der SQL-Ansicht an. Erläutern, wie sich Projektion, Selektion und Join mithilfe eines SQL-Befehls durchführen lassen.

Lösungen

Aufgabe 4:

- a) SELECT DISTINCTROW Buchtyp.Thema, Buchtyp.Titel, Buchtyp.Verlag
FROM Buchtyp
WHERE (((Buchtyp.Thema) Like "Programmieren" & "*") AND
((Buchtyp.Erscheinungsjahr)>1995)) OR (((Buchtyp.Thema) Like
"Programmieren" & "*") AND ((Buchtyp.Verlag)="Microsoft"))
ORDER BY Buchtyp.Titel;
- b) SELECT DISTINCTROW Autor.Autor, Buchtyp.Titel, Buchtyp.Erscheinungsjahr
FROM Buchtyp INNER JOIN Autor ON Buchtyp.[Buchtyp-Nr] = Autor.[Buchtyp-Nr]
WHERE (((Autor.Autor) Like [Autor eingeben] & "*") AND
((Buchtyp.Erscheinungsjahr)>1985 And (Buchtyp.Erscheinungsjahr)<1992))
ORDER BY Autor.Autor, Buchtyp.Titel, Buchtyp.Erscheinungsjahr;
- c) SELECT DISTINCTROW Stichwort.Stichwort, Buchtyp.Titel, Autor.Autor
FROM Stichwort INNER JOIN ((Buchtyp INNER JOIN Autor ON
Buchtyp.[Buchtyp-Nr] = Autor.[Buchtyp-Nr]) INNER JOIN gehört_zu ON
Buchtyp.[Buchtyp-Nr] = gehört_zu.[Buchtyp-Nr]) ON
Stichwort.[Stichwort-Nr] = gehört_zu.[Stichwort-Nr]
WHERE (((Stichwort.Stichwort) Like "*" & [Stichwort eingeben] & "*"));
- d) SELECT Ausleiher.Name, Autor.Autor, Buchtyp.Titel, Ausleihe.Ausleihdatum
FROM ((Buchtyp INNER JOIN Autor ON Buchtyp.[Buchtyp-Nr] =
Autor.[Buchtyp-Nr]) INNER JOIN Ausleihbuch ON Buchtyp.[Buchtyp-Nr] =
Ausleihbuch.[Buchtyp-Nr]) INNER JOIN (Ausleiher INNER JOIN Ausleihe
ON Ausleiher.[Ausleiher-Nr] = Ausleihe.[Ausleiher-Nr]) ON
Ausleihbuch.[Inventar-Nr] = Ausleihe.[Inventar-Nr]
WHERE (((Ausleiher.Name) Like "*" & [Name des Schülers] & "*"));
- e) SELECT DISTINCTROW Ausleiher.Name, Ausleiher.Vorname,
Ausleihe.Ausleihdatum, Ausleihbuch.Ausleihzeit
FROM Ausleihbuch INNER JOIN (Ausleiher INNER JOIN Ausleihe ON
Ausleiher.[Ausleiher-Nr] = Ausleihe.[Ausleiher-Nr]) ON
Ausleihbuch.[Inventar-Nr] = Ausleihe.[Inventar-Nr]
WHERE (((Ausleihe.Ausleihdatum)<#11/30/2006#-[Ausleihbuch].[Ausleihzeit]));
- f) SELECT Buchtyp.Titel, Buchtyp.Erscheinungsjahr, Buchtyp.Verlag,
Count(Ausleihbuch.[Inventar-Nr]) AS [Anzahl von Inventar-Nr]
FROM Buchtyp LEFT JOIN Ausleihbuch ON Buchtyp.[Buchtyp-Nr] =
Ausleihbuch.[Buchtyp-Nr]
GROUP BY Buchtyp.Titel, Buchtyp.Erscheinungsjahr, Buchtyp.Verlag;)
- g) SELECT Ausleihe.[Ausleiher-Nr], Ausleihe.[Inventar-Nr], Ausleihe.Ausleihdatum
INTO AlteAusleihe
FROM Ausleihe
WHERE (((Ausleihe.Ausleihdatum)>#3/15/2005#));
- h) DELETE AlteAusleihe.*
FROM AlteAusleihe;