* **Erzeugen Sie zuerst einen neuen Benutzer namens projekt – tun Sie dies als der Benutzer System. Geben Sie ihm das Passwort projekt – der Befehl lautet:**  
    
  **CREATE USER projekt11**

**IDENTIFIED BY projekt11**

**DEFAULT TABLESPACE users**

**TEMPORARY TABLESPACE temp**

**QUOTA 20M ON users;**

* **Weisen Sie ihm anschließend die Rolle Orax zu – sie beinhaltet alle Rechte, die der User für die folgenden aufgaben braucht. Wie lautet der Befehl?**

CREATE ROLE Orax;

GRANT CREATE SESSION TO Orax;

GRANT CREATE TABLE TO Orax;

GRANT CREATE SEQUENCE TO Orax;

GRANT CREATE VIEW TO Orax;

GRANT Orax TO projekt11;

* **Betrachten Sie das nachfolgende ER-Diagramm – Sie sollen für die Werksbücherei die abgebildete Anwendung erzeugen**



* **Erstellen Sie als der neue Benutzer Projekt zunächst die 7 Tabellen für das abgebildete ERD wie in den nachfolgenden Tabellen abgebildet – lassen Sie die Fremdschlüssel noch weg – Wie lauten die Befehle?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabellenname** | **Mitglieder** |  |  |
| Spalte | Datentyp | Constraint | Abhängigkeiten |
| Memberid | Number(6,0) | PK |  |
| Vorname | Varchar2(25) | Not null |  |
| Nachname | Varchar2(25) | Not null |  |
| Strasse | Varchar2(40) |  |  |
| Plz | Varchar2(5) |  |  |
| Ort | Varchar2(40) |  |  |
|  |  |  |  |
|  |  |  |  |
| **Tabellenname** | **Buecher** |  |  |
| Bookid | Number(8) | PK |  |
| Titel | Varchar2(100) | Not null |  |
|  |  |  |  |
|  |  |  |  |
| **Tabellenname** | **Autoren** |  |  |
| Autorid | Number(8) | PK |  |
| Vorname | Varchar2(25) |  |  |
| Nachname | Varchar2(35) | Not null |  |
|  |  |  |  |
|  |  |  |  |
| **Tabellenname** | **Verlage** |  |  |
| Verlagsid | Number(8) | PK |  |
| Verlagsname | Varchar2(100) | Not null |  |
|  |  |  |  |
|  |  |  |  |
| **Tabellenname** | **Verleihvorgaenge** |  |  |
| Verleihnr | Number(8) | PK |  |
| Verleihdatum | Date |  |  |
| Bookid | Number(8) | FK | FK zu Buecher |
| Memberid | Number(8) | FK | FK zu Mitglieder |
|  |  |  |  |
|  |  |  |  |
| die beiden Tabellen Buch\_zu\_Autor und Buch\_zu\_Verlag besitzen einen Mehrfelderprimärschlüssel und gleichzeitig sind die beiden Spalten einzeln ein Fremdschlüssel | | | |
|  |  | FKs noch nicht erstellen |  |
| **Tabellenname** | **Buch\_zu\_Autor** |  |  |
| Bookid | Number(8) | PK | Buecher |
| Autorid | Number(8) | PK | Autoren |
|  |  |  |  |
| **Tabellenname** | **Buch\_zu\_Verlag** |  |  |
| Bookid | Number(8) | PK | Buecher |
| Verlagsid | Number(8) | PK | Verlage |
|  |  |  |  |

CREATE TABLE Mitglieder

(

Memberid Number(6,0) constraint pk\_mitglieder primary key,

Vorname Varchar2(25) Not Null,

Nachname Varchar2(25) Not Null,

Strasse Varchar2(40),

Plz Varchar2(5),

Ort Varchar2(40)

);

desc Mitglieder ;

CREATE TABLE Buecher

(

Bookid Number(8) constraint pk\_buecher primary key,

Titel Varchar2(40) Not Null

);

desc Buecher;

CREATE TABLE Autoren

(

Autorid Number(8) constraint pk\_autoren primary key,

Vorname Varchar2(25),

Nachname Varchar2(35) Not Null

);

desc Autoren;

CREATE TABLE Verlage

(

Verlagsid Number(8) constraint pk\_verlage primary key,

Verlagsname Varchar2(40) Not Null

);

desc Verlage;

CREATE TABLE Verleihvorgaenge

(

Verleihnr Number(8) constraint pk\_verleihvorgaenge primary key,

Verleihdatum Date,

Bookid Number(8),

Memberid Number(8)

);

desc Verleihvorgaenge;

CREATE TABLE Buch\_zu\_Autor

(

Bookid Number(8) ,

Autorid Number(8) ,

constraint pk\_buch\_zu\_autor primary key (bookid,autorid)

);

desc Buch\_zu\_Autor;

CREATE TABLE Buch\_zu\_Verlag

(

Bookid Number(8),

Verlagsid Number(8),

constraint pk\_buch\_zu\_verlag primary key (bookid, verlagsid)

);

desc Buch\_zu\_Verlag;

* **Jetzt erzeugen Sie eine Sequenz für die Mitgliedernr der Tabelle Mitglieder. Der Name der Sequenz soll Mitgliedseq lauten, der Startwert soll 201 sein und jeweils um 1 hochgezählt werden – kein Caching erlaubt. Wie lautet die Anweisung?**

CREATE SEQUENCE mitgliedseq

START WITH 201

INCREMENT BY 1

NOCACHE;

* **Öffnen Sie nun im SQLdeveloper die Datei, die die Anweisungen zur Dateneingabe in die Tabelle Mitglieder enthält und führen Sie die Befehle aus. Sie heißt: Mitgliederdaten einfügen.sql. Führen Sie das Skript aus.**

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Maria','Anders','Obere strasse. 57','12209','Berlin');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Hanna','Moos','Forsterstr. 57','68306','Mannheim');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Sven','Ottlieb','Walserweg 21','52520','Aachen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Peter','Franken','Berliner Platz 43','80805','Muenchen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Philip','Cramer','Maubelstr. 90','14776','Brandenburg');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Renate','Messner','Magazinweg 7','60528','Frankfurt a. M.');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Alexander','Feuer','Heerstr 22','04179','Leipzig');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Henriette','Pfalzheim','Mehrheimerstr. 369','50739','Koeln');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Horst','Kloos','Taucherstraße 10','01307','Cunewalde');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Karin','Josephs','Luisenstr. 48','44087','Muenster');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Rita','Mueller','Danzinger Straße 12','71073','Stuttgart');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Herbert','Mueller','Kraehenweg 4','81347','Muenchen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Karl','Friesen','Rennweg 23a','70934','Lauben');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Helmut','Werner','Hauptstraße 14','70883','Metzingen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Renate','Sauer','Maubeerstr. 245','32870','Kleineich');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Josef','Mueller','Westlicher Weg 14','80510','Muenchen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Vanessa','Geiger','Amselweg 4','81309','Muenchen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Sigrid','Kanter','Forellenstr. 17','60351','Frankfurt a.M.');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Franz','Maler','Hauptstr. 53','93041','Hof');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Gerhard','Reiter','Braunstr. 14','82561','Großmanching');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Karl','Fischer','Strandweg 2','20034','Hamburg');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Maria','Kleinert','Weizenstr. 67','11452','Berlin');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Karl','Jaeger','Alsterstr. 89','20851','Hamburg');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Siegfried','Meister','Auenstr. 145','11743','Berlin');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Adalbert','Koenig','Weiherweg 3','21409','Hamburg');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Rudolf','Plattner','Baerenstr. 67','61308','Frankfurt a.M.');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Herbert','Frenzen','Obermuehle 4','93481','Merten');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Heinz','Hansen','Hochstr. 35','60411','Frankfurt a.M.');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Renate','Schnell','Gaensestr. 23','61423','Melsen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Reinhard','Schimmel','Hammelweg 14','60432','Frankfurt a.M.');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Lieschen','Mueller','Neue Gasse','79990','Stuttgart');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Rudi','Siegel','Allee 9','70467','Stuttgart');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Anita','Bücher','Momserweg 2','44368','Norderstedt');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Otto','Reichert','Arkadenstr. 14','93688','Würzlingen');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Fred','Feuerstein','Geröllhalde 20','12345','Geröllheim');

Insert into mitglieder (memberid,VORNAME,nachname,strasse,PLZ,ORT)

values (mitgliedseq.nextval,'Barny','Geröllheimer','Geröllhalde 20','12345','Geröllheim');

commit;

select \* from mitglieder;

* **Im nächsten Schritt fügen Sie die Daten für Autoren und Verlage ein – Sie finden diese in der Datei Autoren und Verlage einfügen.sql ebenfalls im Verzeichnis Projekt – öffnen Sie die Datei und lassen Sie die Inserts ausführen – damit befüllen Sie die Tabelle Autoren und Verlage mit Daten.**

INSERT INTO autoren(autorid, vorname, nachname)values ('92','George Bernard','Shaw');

INSERT INTO autoren(autorid, vorname, nachname)VALUES ('93','Leo','Tolstoy');

INSERT INTO autoren(autorid, vorname, nachname) values ('94','Percy', 'Shelley');

INSERT INTO autoren(autorid, vorname, nachname)VALUES ('95','Lord', 'Byron');

INSERT INTO autoren(autorid, vorname, nachname) values ('96','John', 'Keats');

INSERT INTO autoren(autorid, vorname, nachname) values ('97','Rudyard', 'Kipling');

INSERT INTO autoren(autorid, vorname, nachname) values ('98','Oliver', 'Goldsmith');

INSERT INTO autoren(autorid, vorname, nachname) values ('99','Oscar', 'Wilde');

INSERT INTO autoren(autorid, vorname, nachname)VALUES ('100','P. G.', 'Wodehouse');

commit;

select \* from autoren;

INSERT INTO verlage(verlagsid, verlagsname) values (100,'Elsevier');

INSERT INTO verlage(verlagsid, verlagsname) values (200,'Penguin Group');

INSERT INTO verlage(verlagsid, verlagsname) values (300,'Pearson Education');

INSERT INTO verlage(verlagsid, verlagsname) values (400,'Cambridge University Press ');

INSERT INTO verlage(verlagsid, verlagsname) values (500,'Dorling Kindersley ');

INSERT INTO verlage(verlagsid, verlagsname) values (301,'AC Barls');

INSERT INTO verlage(verlagsid, verlagsname) values (302,'Penguin Group');

INSERT INTO verlage(verlagsid, verlagsname) values (303,'Elsevier\_II');

INSERT INTO verlage(verlagsid, verlagsname) values (304,'Pearson Education');

INSERT INTO verlage(verlagsid, verlagsname) values (305,'Inc Publication');

INSERT INTO verlage(verlagsid, verlagsname) values (306,'Embel');

INSERT INTO verlage(verlagsid, verlagsname) values (307,'Anthoys');

INSERT INTO verlage(verlagsid, verlagsname) values (308,'Cengage');

INSERT INTO verlage(verlagsid, verlagsname) values (309,'Wilsey');

INSERT INTO verlage(verlagsid, verlagsname) values (310,'Mc Graw');

commit;

select \* from verlage;

* **Erstellen Sie eine weitere Sequenz, diesmal für die Autorid der Tabelle Autoren - lassen Sie die Sequenz mit der Nummer 101 beginnen (es gibt ja bereits Autoren mit Autorids – deshalb starten Sie hier nicht mit 1), Inkrement soll 1 sein, es sollen keine Nummern gecached werden – Wie lautet die Anweisung?**

CREATE SEQUENCE autorenseq

START WITH 101

INCREMENT BY 1

NOCACHE;

* **fügen Sie danach noch folgende Autoren hinzu - verwenden Sie die neue Sequenz und die Austauschvariable & für die Eingabe – wie muss der Befehl lauten, den Sie dann achtmal ausführen?**

|  |  |  |
| --- | --- | --- |
| Autorid | Vorname | Nachname |
| 101 | Thomas | Hardy |
| 102 |  | Shakespeare |
| 103 | H.G. | Wells |
| 104 | Rudyard | Kipling |
| 105 |  | Kalidas |
| 106 | John | Milton |
| 107 | R.K. | Narayan |
| 108 | Charles | Dickens |

Insert into autoren (autorid,VORNAME,nachname) values (autorenseq.nextval,'&Vorname','&Nachname');

commit;

select \* from autoren;

* **Jetzt öffnen Sie die Datei Buecherdaten einfügen.sql und lassen die inserts ausführen**

INSERT INTO buecher values (1,'Florentine Tragedy');

INSERT INTO buecher values (2,'A Vision');

INSERT INTO buecher values (3,'Citizen of the World');

INSERT INTO buecher values (4,'The Complete Poetical Works of Oliver');

INSERT INTO buecher values (5,'Androcles and the Lion');

INSERT INTO buecher values (6,'An Unsocial Socialist');

INSERT INTO buecher values (7,'A Thing of Beauty is a Joy Forever');

INSERT INTO buecher values (8,'Beyond the Pale');

INSERT INTO buecher values (9,'The Clicking of Cuthbert');

INSERT INTO buecher values (10,'Bride of Frankenstein');

INSERT INTO buecher values (11,'Shelley Poetry and Prose');

INSERT INTO buecher values (12,'War and Peace');

INSERT INTO buecher values (13,'Invisible Man');

Insert into buecher values(14, 'The Girl with the dragon Tattoo');

Insert into buecher values(15, 'Raghuvamsa');

Insert into buecher values(16, 'Shakuntala');

Insert into buecher values(17, 'Antony and Cleopatra');

Insert into buecher values(18, 'Oliver Twist');

Insert into buecher values(19, 'Introduction to Database');

Insert into buecher values(20, 'Jungle Book');

Insert into buecher values(21, 'Paradise lost');

Insert into buecher values(22, 'Othello');

Insert into buecher values(23, 'Pickwick Papers');

Insert into buecher values(24, 'Tempest');

commit;

select \* from buecher;

1. Nun befüllen Sie die beiden Tabellen, die die 2 many-to-many Beziehungen zwischen Bücher und Autoren sowie Bücher und Verlagen herstellen

Insert into buch\_zu\_autor(autorid, bookid) values(103,13);

Insert into buch\_zu\_autor(autorid, bookid) values(101,14);

Insert into buch\_zu\_autor(autorid, bookid) values(105,15);

Insert into buch\_zu\_autor(autorid, bookid) values(105,16);

Insert into buch\_zu\_autor(autorid, bookid) values(102, 17);

Insert into buch\_zu\_autor(autorid, bookid) values(108, 18);

Insert into buch\_zu\_autor(autorid, bookid) values(107, 19);

Insert into buch\_zu\_autor(autorid, bookid) values(104, 20);

Insert into buch\_zu\_autor(autorid, bookid) values(106, 21);

Insert into buch\_zu\_autor(autorid, bookid) values(102, 22);

Insert into buch\_zu\_autor(autorid, bookid) values(108, 23);

Insert into buch\_zu\_autor(autorid, bookid) values(102,24);

insert into buch\_zu\_autor(autorid, bookid) values(93, 1);

insert into buch\_zu\_autor(autorid, bookid) values(93,2);

insert into buch\_zu\_autor(autorid, bookid) values(92, 3);

insert into buch\_zu\_autor(autorid, bookid) values(92, 4);

insert into buch\_zu\_autor(autorid, bookid) values(94, 5);

insert into buch\_zu\_autor(autorid, bookid) values(94, 6);

insert into buch\_zu\_autor(autorid, bookid) values(98, 7);

insert into buch\_zu\_autor(autorid, bookid) values(96, 8);

insert into buch\_zu\_autor(autorid, bookid) values(98, 9);

insert into buch\_zu\_autor(autorid, bookid) values(96, 10);

insert into buch\_zu\_autor(autorid, bookid) values(95, 11);

insert into buch\_zu\_autor(autorid, bookid) values(94, 12);

commit;

select \* from buch\_zu\_autor;

insert into buch\_zu\_verlag(bookid, verlagsid) values(1,200);

insert into buch\_zu\_verlag(bookid, verlagsid) values(2,300);

insert into buch\_zu\_verlag(bookid, verlagsid) values(3,100);

insert into buch\_zu\_verlag(bookid, verlagsid) values(4,100);

insert into buch\_zu\_verlag(bookid, verlagsid) values(5,400);

insert into buch\_zu\_verlag(bookid, verlagsid) values(6,400);

insert into buch\_zu\_verlag(bookid, verlagsid) values(7,200);

insert into buch\_zu\_verlag(bookid, verlagsid) values(8,500);

insert into buch\_zu\_verlag(bookid, verlagsid) values(9,500);

insert into buch\_zu\_verlag(bookid, verlagsid) values(10,100);

insert into buch\_zu\_verlag(bookid, verlagsid) values(11,300);

insert into buch\_zu\_verlag(bookid, verlagsid) values(12,200);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(13, 301);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(15,303);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(15,303);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(16, 303);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(17, 306);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(18, 305);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(19, 304);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(20, 307);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(21, 308);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(22, 304);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(23, 305);

Insert into buch\_zu\_verlag(bookid, verlagsid) values(24,304);

commit;

select \* from buch\_zu\_verlag;

Insert into VERLEIHVORGAENGE values (501,'20.04.15',1,201);

Insert into VERLEIHVORGAENGE values (502, '05.04.15',2,203);

Insert into VERLEIHVORGAENGE values (503,'15.12.15',11,202);

Insert into VERLEIHVORGAENGE values (504,'25.10.15',4,204);

Insert into VERLEIHVORGAENGE values (505,'13.02.16',5,212);

Insert into VERLEIHVORGAENGE values (506,'12.08.15',4,209);

Insert into VERLEIHVORGAENGE values (507,'10.03.15',12,210);

Insert into VERLEIHVORGAENGE values (508,'07.05.15',8,205);

Insert into VERLEIHVORGAENGE values (509,'02.01.15',10,209);

Insert into VERLEIHVORGAENGE values (510,'15.01.15',11,210);

Insert into VERLEIHVORGAENGE values (511,'01.04.16',10,215);

Insert into VERLEIHVORGAENGE values (512,'29.02.16',5,213);

Insert into VERLEIHVORGAENGE values (513,'13.12.15',9,207);

Insert into VERLEIHVORGAENGE values (514,'28.11.15',5,205);

Insert into VERLEIHVORGAENGE values (515,'03.01.16',2,214);

Insert into VERLEIHVORGAENGE values (516,'21.10.15',3,215);

Insert into VERLEIHVORGAENGE values (517,'14.09.17',14,213);

Insert into VERLEIHVORGAENGE values (518,'14.01.18',23,202);

Insert into VERLEIHVORGAENGE values (519,'18.09.19',11,205);

Insert into VERLEIHVORGAENGE values (520,'18.03.20',15,218);

Insert into VERLEIHVORGAENGE values (521,'10.11.20',9,220);

Insert into VERLEIHVORGAENGE values (522,'14.07.20',23,202);

Insert into VERLEIHVORGAENGE values (523,'15.05.19',22,207);

Insert into VERLEIHVORGAENGE values (524,'14.06.18',21,202);

commit;

select \* from VERLEIHVORGAENGE;

* **Lassen Sie sich die Daten aller Tabellen einmal anzeigen mit Select – klicken Sie links im Strukturfenster auf die Tabelle buch\_zu\_Autor und lassen Sie sich das Modell anzeigen. Machen Sie das gleiche mit Buch\_zu\_Verlag und mit den Verleihvorgaengen – dies sollte Ihnen einen Überblick über die erstellten Tabellen und deren ERD geben.**

SELECT \* FROM mitglieder;

SELECT \* FROM Buecher;

SELECT \* FROM Autoren;

SELECT \* FROM Verlage;

SELECT \* FROM Verleihvorgaenge;

SELECT \* FROM buch\_zu\_autor;

SELECT \* FROM buch\_zu\_verlag;

* **Jetzt fügen Sie die Fremdschlüssel hinzu, für jeden Constraint 1 Befehl und zwar wie folgt:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Von Tabelle** | **spalte** | **Nach Tabelle** | **Spalte** | **Constrainttyp** |
| Verleihvorgänge | Bookid | Buecher | bookid | Foreign key |
| Verleihvorgänge | Memberid | Mitglieder | Memberid | Foreign key |
| Buch\_zu\_Autor | Bookid | Buecher | Bookid | Foreign key |
| Buch\_zu\_Autor | Autorid | Autoren | Autorid | Foreign key |
| Buch\_zu\_Verlag | Bookid | Buecher | Bookid | Foreign key |
| Buch\_zu\_Verlag | Verlagsid | Verlage | Verlagsid | Foreign key |
|  |  |  |  |  |

ALTER TABLE Verleihvorgaenge

ADD CONSTRAINT FK\_Verleihvorgaenge

FOREIGN KEY (bookid) REFERENCES Buecher (bookid);

ALTER TABLE Verleihvorgaenge

ADD CONSTRAINT FK\_Verleihvorgaenge\_2

FOREIGN KEY (memberid) REFERENCES mitglieder (memberid);

ALTER TABLE buch\_zu\_autor

ADD CONSTRAINT FK\_buch\_zu\_autor

FOREIGN KEY (bookid) REFERENCES buecher (bookid);

ALTER TABLE buch\_zu\_autor

ADD CONSTRAINT FK\_buch\_zu\_autor\_2

FOREIGN KEY (autorid) REFERENCES autoren (autorid);

ALTER TABLE buch\_zu\_verlag

ADD CONSTRAINT FK\_buch\_zu\_verlag

FOREIGN KEY (bookid) REFERENCES buecher (Bookid);

ALTER TABLE buch\_zu\_verlag

ADD CONSTRAINT FK\_buch\_zu\_verlag\_2

FOREIGN KEY (verlagsid) REFERENCES verlage (verlagsid);

* **Erzeugen Sie eine View namens MitgliederDetails, diese soll Ihnen den Namen, Strasse, Plz und Ort von Mitgliedern anzeigen sowie den Titel der Buecher, die diese entliehen haben.**

CREATE VIEW MitgliederDetails

AS

SELECT

m.Vorname, m.Nachname, m.Strasse, m. Plz, m.Ort, b.Titel Titel\_der\_Buecher

from mitglieder m

INNER JOIN Verleihvorgaenge v

on (m.memberid = v.memberid)

INNER JOIN Buecher b

on (v.bookid = b.bookid);

select \* from MitgliederDetails;

1. Erweitern Sie die Daten in der Buechertabelle wie folgt:  
   Fügen Sie ein neues Buch hinzu, prüfen Sie vorher, ob es den Autor bzw. den genannten Verlag schon gibt, wenn nicht geben Sie die erforderlichen Daten in die entsprechenden Tabellen ein. In welchen Tabellen muss überall ein Eintrag erfolgen? Wie lauten die Einträge?

|  |  |  |
| --- | --- | --- |
| Titel | Verlag | Autor |
| Great Expectations | Inc Publication | Charles Dickens |

SELECT \* FROM buecher

WHERE titel = 'Great Expectations';

SELECT \* FROM Verlage

WHERE verlagsname= 'Inc Publication';

SELECT \* FROM Autoren

WHERE vorname ='Charles';

select\* from buecher;

CREATE SEQUENCE bookidseq

START WITH 25

INCREMENT BY 1

NOCACHE;

INSERT INTO buecher VALUES(bookidseq.nextval, 'Great Expectations') ;

INSERT INTO buch\_zu\_autor VALUES ((select max(bookid) from buecher), (select autorid from autoren where nachname='Dickens')) ;

INSERT INTO buch\_zu\_verlag(bookid,verlagsid) VALUES((select max(bookid) from buecher), (select verlagsid from verlage where verlagsname='Inc Publication'));

commit;

* **Sven Ottlieb leiht das Buch Jungle book aus der Bücherei – welche Transaktion ist erforderlich um den Verleihvorgang festzuhalten?**

SELECT memberid FROM Mitglieder where vorname ='Sven' and Nachname='Ottlieb';

SELECT bookid FROM buecher where titel = 'Jungle Book';

CREATE SEQUENCE Verleihvorgaengeseq

START WITH 525

INCREMENT BY 1

NOCACHE;

INSERT INTO Verleihvorgaenge VALUES (Verleihvorgaengeseq.nextval, sysdate, (SELECT bookid FROM buecher where titel = 'Jungle Book'), (SELECT memberid FROM Mitglieder where vorname ='Sven' and Nachname='Ottlieb'));

commit;

select \* from Verleihvorgaenge;

* **Schreiben Sie eine Abfrage, die die Namen aller Mitglieder ermittelt, die das Buch Shakuntala jemals ausgliehen haben**

SELECT m.vorname, m.nachname, b.titel, v.verleihnr, v.verleihdatum from mitglieder m Join verleihvorgaenge v on (v.memberid=m.memberid)

join buecher b on (b.bookid=v.bookid) where b.titel = 'Shakuntala'

select \* from verleihvorgaenge;

* **Schreiben Sie eine Abfrage, die alle Bücher von Shakespeare findet, die in der Bücherei vorhanden sind.**

select a.nachname Nachname\_Autor, b.titel, b.bookid, bza.autorid from autoren a

left join buch\_zu\_autor bza

on (bza.autorid = a.autorid)

left join buecher b

on (b.bookid=bza.bookid)

where a.nachname = 'Shakespeare';

* **Einige Bücher sind so häufig nachgefragt, dass die Entscheidung getroffen wurde, dass mehrere Exemplare gekauft werden sollen. Fügen Sie der Buechertabelle eine neue Spalte Anzahl\_Kopien hinzu. Die Spalte soll eine Zahl enthalten wie in der unten aufgeführten Hilfstabelle dargestellt.** 
  1. **Wie lautet der Befehl um die Spalte zu erzeugen?**

**Erzeugen Sie nun die nachfolged dargestellte Hilfstabelle, geben Sie die Daten ein.**

|  |  |
| --- | --- |
| **Hilfstabelle** |  |
| Bookid | Anzahl\_Kopien |
| 1 | 10 |
| 2 | 6 |
| 3 | 4 |
| 4 | 3 |
| 5 | 8 |
| 6 | 9 |
| 7 | 9 |
| 8 | 2 |
| 9 | 7 |
| 10 | 5 |
| 11 | 2 |
| 12 | 4 |
| 13 | 6 |
| 14 | 3 |
| 15 | 8 |
| 16 | 2 |
| 17 | 3 |
| 18 | 3 |
| 19 | 2 |
| 20 | 4 |
| 21 | 9 |
| 22 | 3 |
| 23 | 2 |
| 24 | 5 |

* **Ändern Sie dann die Buechertabelle mit den Daten aus der Spalte Anzahl\_Kopien in der Hilfstabelle für vorhandene Bücher – verwenden Sie eine korrelierende Update-Anweisung um die leere Spalte in der Buechertabelle mit den Daten zu befüllen.**

ALTER TABLE buecher

ADD Anzahl\_Kopien number(2);

create table hilfstabelle

(

bookid number(6,0) constraint pk\_hilfstabelle primary key,

Anzahl\_Kopien number(2)

);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (1, 10);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (2, 6);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (3, 4);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (4, 3);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (5, 8);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (6, 9);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (7, 9);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (8, 2);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (9, 7);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (10,5);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (11,2);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (12,4);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (13,6);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (14,3);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (15,8);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (16,2);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (17,3);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (18,3);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (19,2);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (20,4);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (21,9);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (22,3);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (23,2);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (24,5);

insert into hilfstabelle (bookid, Anzahl\_Kopien) values (25,5);

commit;

select \* from hilfstabelle;

UPDATE buecher b SET Anzahl\_Kopien = (select Anzahl\_Kopien

FROM hilfstabelle h

WHERE h.bookid = b.bookid);

select \* from buecher;

* **Erzeugen Sie eine View namens Nachfrage, die alle Buchtitel anzeigt, auch die, die noch nie verliehen wurden, den dazugehörigen Autor und Verlag, den Namen des Mitglieds, der das Buch geliehen hat, das Datum, wann das Buch verliehen wurde – bei allen Büchern, die noch nie verliehen wurden soll in den Feldern, die dadurch leer bleiben 2 (--) aufeinanderfolgende Minuszeichen ausgegeben werden**

CREATE OR REPLACE VIEW Nachfrage

AS

SELECT b.titel, a.Vorname Vorname\_autor, a.Nachname Nachname\_autor, NVL(TO\_CHAR(v.verleihdatum), '--') Verleihdatum, NVL(m.vorname,'--') Vorname\_mitglieder, NVL(m.nachname,'--') Nachname\_mitglieder, v.Verlagsname Verlagsname

FROM buecher b LEFT JOIN verleihvorgaenge v on (b.bookid = v.bookid)

LEFT OUTER JOIN mitglieder m on (m.memberid = v.memberid)

LEFT JOIN buch\_zu\_verlag bzv on (b.bookid=bzv.bookid)

LEFT JOIN verlage v on (bzv.verlagsid= v.verlagsid)

LEFT JOIN buch\_zu\_autor bza on (bza.bookid = b.bookid)

LEFT OUTER JOIN autoren a on (bza.autorid = a.autorid)

select \* from nachfrage;

* **Erzeugen Sie eine View Namens Never\_Read – lassen Sie alle Mitglieder (Vorname, Nachname und memberid) ausgeben, die noch nie ein Buch entliehen haben – verwenden Sie für die Lösung einen Set Operator (Mengenoperator). Ermitteln Sie über eine Selectanweisung wieviele Mitglieder das sind. Notieren Sie die Zahl – beachten Sie, dass Mitglieder nicht mehrfach gezählt werden dürfen.**

CREATE OR REPLACE VIEW Never\_Read

AS

SELECT m.Vorname, m.Nachname, m.Memberid,NVL(TO\_CHAR( v.Verleihdatum),'--') Verleihdatum

FROM Mitglieder m LEFT JOIN verleihvorgaenge v on (v.memberid = m.memberid)

INTERSECT

SELECT m.Vorname, m.Nachname, m.Memberid, NVL(TO\_CHAR( v.Verleihdatum),'--') Verleihdatum

FROM Mitglieder m LEFT JOIN verleihvorgaenge v on (v.memberid = m.memberid) where v.verleihdatum is NULL

select count(memberid) from Never\_Read; -- 22 Mitglieder

* **Und jetzt noch eine View Reader, die alle ausgibt (Vorname, Nachname, Buchtitel, memberid), die schon einmal ein Buch entliehen haben. Erzeugen sie auch hier eine Selectanweisung auf die View, die zählt wieviele Mitglieder dies sind. Notieren Sie die Zahl**

CREATE OR REPLACE VIEW Reader

AS

SELECT m.Vorname, m.Nachname, m.Memberid, NVL(TO\_CHAR( v.Verleihdatum),'--') Verleihdatum, b.titel

FROM Mitglieder m LEFT JOIN verleihvorgaenge v on (v.memberid = m.memberid)

LEFT JOIN buecher b on (b.bookid = v.bookid)

where v.verleihdatum is NOT NULL

SELECT COUNT(DISTINCT memberid) FROM Reader; --14 Mitglieder

* **Es finden folgende Verleihvorgänge statt – führen Sie bitte die notwendigen Befehle hierfür aus:**  
  Renate Messner leiht das Buch Citizen of the World  
  Henriette Pfalzheim leiht das Buch Bride of Frankenstein  
  Rita Mueller leiht das Buch Invisible Man  
  Vanessa Geiger leiht das Buch Shakuntala  
  Maria Kleinert leiht das Buch Oliver Twist

select \* from verleihvorgaenge;

CREATE SEQUENCE verleihidseq

start with 526

increment by 1

maxvalue 700

nocache;

Insert into VERLEIHVORGAENGE

values (verleihidseq.nextval, SYSDATE, (SELECT DISTINCT b.bookid from buecher b FULL OUTER JOIN verleihvorgaenge v on (b.bookid=v.bookid) where b.titel = '&Buchtitel'),

(SELECT memberid from mitglieder b where vorname ='&Vorname' and nachname ='&Nachname'));

select \* from verleihvorgaenge;

* **Fragen Sie das Data Dictionary nach allen Tabellen ab, die Ihnen gehören – wie lautet der Befehl?**

select \* from user\_tables;

* **Fragen Sie außerdem alle Ihre Sequenzen ab – Befehl?**

select \* from user\_sequences;

* **Erzeugen Sie eine View Buecher\_Liste, die den Titel, den Autor und die Anzahl der Kopien aller Bücher ausgibt.**

CREATE OR REPLACE VIEW Buecher\_Liste

AS

SELECT b.titel, a.vorname, a.nachname, b.Anzahl\_Kopien

FROM Buecher b LEFT JOIN buch\_zu\_Autor bza

on (b.bookid = bza.bookid)

left join Autoren a

on (bza.autorid = a.autorid)

order by 1;

select \* from buecher\_liste;

* **Sie erzeugen einen neuen Benutzer namens login\_books, geben Sie ihm das passwort books – weisen sie ihm die Rolle orax zu – gewähren Sie ihm Leserechte auf die View Buecher\_Liste  
    
  Erzeugen Sie eine neue Verbindung für login\_books im SQLdeveloper  
    
  Testen Sie, ob Sie die View Buecher\_liste lesen können.**

CREATE USER login\_books

IDENTIFIED BY books;

connect login\_books/books;

GRANT Orax to login\_books;

GRANT SELECT ON projekt11.Buecher\_Liste to Orax;

--(in login\_books)