

Data-Mart-Erstellung in SQL

Erarbeitung-/Reflexionsphase

Markus Wohlgenannt

August 20, 2024

Einleitung

In der ersten Phase der Konzeptionsphase wurde die Spezifikation und das dazugehörige ERM Modell erstellt. Die einzelnen Attribute wurden in einem Datenwörterbuch beschrieben. In dieser Phase wurden alle Überlegungen theoretisch durchgeführt. Mit dem Übergang der Konzeptionsphase in die Implementierungsphase beginnt nun der praktische Teil des Projekts. Die in der Konzeptionsphase erarbeiteten theoretischen Konzepte müssen in eine Datenbank überführt werden. Dies erfordert eine präzise Umsetzung des ERM in ein relationales Datenbanksystem, das die Grundlage für die Buchtausch-App bildet.

Erstellung der Datenbank

Zunächst wurde eine relationale Datenbank auf Basis eines MySQL-Systems erstellt. Für jede Tabelle musste ein SQL-Skript erstellt werden das die jeweilige Tabelle in der Datenbank nach der Vorlage im ERM anlegt. Jede Tabelle wurde im Anschluss mit Testdaten gefüllt.

Während der Implementierung stellten wir fest, dass die Tabelle `tbl_locations` redundant war. Diese Tabelle war ursprünglich vorgesehen, um verschiedene Standorte für die Bücher zu speichern. Nach eingehender Analyse wurde jedoch klar, dass die Funktionalität dieser Tabelle durch andere bereits existierende Tabellen `tbl_boxes` und `tbl_addresses` abgedeckt wird. Die `tbl_boxes` speichert Informationen über die physischen Boxen, in denen Bücher gelagert werden können, während `tbl_addresses` die spezifischen Adressen dieser Boxen enthält. Diese beiden Tabellen zusammen bieten eine vollständige Abdeckung der Standortinformationen, die ursprünglich für `tbl_locations` vorgesehen waren. Die restlichen Tabellen wurden wie in der Spezifikation beschrieben angelegt.

Änderungen bei der Implementierung

Beim durchführen der Tests merkten wir dass die Wunschliste `tbl_wishlist` nicht ohne weiteres befüllt werden kann. Es kann kein Buch angelegt werden wenn es kein Besitzer hat. Um die Datenintegrität zu gewährleisten und sicherzustellen, dass jedes Buch einem Benutzer zugeordnet ist, haben wir beschlossen, einen "Dummy Wishlist" Benutzer einzuführen. Diese Maßnahme ermöglicht es uns, die bestehenden Schlüsselbeziehungen zwischen den Tabellen beizubehalten. Der Hauptfokus

der App liegt auf dem Verleihen und Ausleihen von Büchern, wobei jedes Buch einen eindeutigen Besitzer haben muss. Durch die Einführung dieses Dummy-Benutzers können wir Änderungen an den Schlüsselbeziehungen vermeiden und gleichzeitig die Konsistenz der Daten sicherstellen. Im Vergleich zu den theoretischen Überlegungen aus der Konzeptionsphase stellten wir beim Befüllen der Testdaten fest, dass ein VARCHAR mit 255 Zeichen sehr großzügig dimensioniert war. Nach erneuter Betrachtung der Testdaten haben wir die Länge bei fast allen Feldern auf 50 Zeichen reduziert. Dies ist ausreichend, um alle notwendigen Informationen zu erfassen und Fehleingaben wie z.B. Abschreibfehler zu vermeiden. Diese Anpassung bietet zudem eine höhere Speichereffizienz und führt zu verbesserten Suchgeschwindigkeiten.

Außerdem wurden Datentypen und Elemente identifiziert, die zunächst falsch interpretiert wurden. Beispielsweise haben Felder wie `postal_code` und `ISBN` immer dieselbe Länge und können daher effizienter als CHAR gespeichert werden.“

Datenbankinstallation

In diesem Abschnitt wird die Installation und Einrichtung der Datenbank für das Projekt "Library Liberty" dokumentiert. Die Datenbankinstallation umfasst die Erstellung der erforderlichen Tabellen und die Konfiguration der Beziehungen zwischen diesen Tabellen gemäß dem zuvor entwickelten Entity-Relationship-Modell (ERM). Dieser Prozess stellt sicher, dass die Datenbank die notwendigen Strukturen und Verbindungen aufweist, um die verschiedenen Funktionen und Anforderungen des Systems zu unterstützen. Darüber hinaus werden exemplarische Testdaten eingefügt, um die Funktionalität der Datenbank zu überprüfen und sicherzustellen, dass sie korrekt arbeitet.

Tabelle: `tbl_addresses`

Zuerst erstellen wir eine Schema in dem wir alle Tabellen für die App anlegen. Die Tabelle `tbl_addresses` speichert die Adressinformationen der Benutzer und der Lagerboxen. Diese Tabelle enthält wichtige Informationen zur Lokalisierung von Benutzern und Büchern. Sie enthält die folgenden Felder:

- `address_id` - Eindeutige ID der Adresse (Primärschlüssel)
- `street` - Straßename und -nummer
- `city` - Stadtname
- `postal_code` - Postleitzahl
- `country` - Land
- `description` - Beschreibung der Adresse (z.B. Wohnadresse, Bücherbox)
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Adresse gelöscht wurde

```

1   -- Creates the Schema for the Tables
2 *  CREATE SCHEMA `dbo`;
3
4  -- Table for storing addresses
5
6 *  CREATE TABLE dbo.tbl_addresses (
7      address_id INT AUTO_INCREMENT PRIMARY KEY,
8      street VARCHAR(50),
9      city VARCHAR(50),
10     postal_code VARCHAR(10),
11     country VARCHAR(50),
12     description VARCHAR(50),
13     deleted BOOLEAN DEFAULT FALSE
14 );
15
16  -- Insert data into dbo.tbl_addresses
17 *  INSERT INTO dbo.tbl_addresses (street, city, postal_code, country, description) VALUES
18     ('Bahnhofstraße 1', 'Bregenz', '6900', 'Österreich', 'Wohnadresse'),
19     ('Dorfstraße 2', 'Dornbirn', '6930', 'Österreich', 'Wohnadresse'),
20     ('Hauptstraße 3', 'Feldkirch', '6800', 'Österreich', 'Wohnadresse'),
21     ('Landstraße 4', 'Bludenz', '6700', 'Österreich', 'Wohnadresse'),
22     ('Obere Hauptstraße 5', 'Hohenems', '6845', 'Österreich', 'Wohnadresse'),
23     ('Kirchplatz 6', 'Hohenems', '6845', 'Österreich', 'Bücherbox'),
24     ('Gartemweg 7', 'Lustenau', '6890', 'Österreich', 'Bücherbox'),
25     ('Alte Straße 8', 'Götzis', '6840', 'Österreich', 'Bücherbox'),
26     ('Hüllerpassage 9', 'Baumgarten', '6830', 'Österreich', 'Bücherbox'),
27     ('Bergstraße 10', 'Wölfling', '6922', 'Österreich', 'Bücherbox'),
28     ('Schulstraße 11', 'Bregenz', '6900', 'Österreich', 'Bücherbox'),
29     ('Parkstraße 12', 'Dornbirn', '6950', 'Österreich', 'Bücherbox'),
30     ('Bibliotheksweg 13', 'Feldkirch', '6800', 'Österreich', 'Bücherbox'),
31     ('Rathausplatz 14', 'Bludenz', '6700', 'Österreich', 'Bücherbox'),
32     ('Museumsstraße 15', 'Hohenems', '6845', 'Österreich', 'Bücherbox');
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
623
624
625
625
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
14
```

Tabelle: `tbl_boxes`

Die Tabelle `tbl_boxes` speichert Informationen zu den Lagerboxen, in denen Bücher aufbewahrt werden können. Diese Tabelle ermöglicht es, die Positionen der Bücher genau zu verfolgen. Sie enthält die folgenden Felder:

- `box_id` - Eindeutige ID der Box (Primärschlüssel)
- `address_id` - Verknüpfte Adresse der Box (Fremdschlüssel)
- `description` - Beschreibung des Standortes der Box
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Box gelöscht wurde

```
1  -- Table for storing book boxes, describes where the Boxes are stored.
2 • CREATE TABLE dbo.tbl_boxes (
3     box_id INT AUTO_INCREMENT PRIMARY KEY,
4     address_id INT,
5     description VARCHAR(255),
6     deleted BOOLEAN DEFAULT FALSE,
7     CONSTRAINT fk_box_address FOREIGN KEY (address_id) REFERENCES dbo.tbl_addresses(address_id)
8 );
9
10 -- Insert data into tbl_boxes
11
12 • INSERT INTO dbo.tbl_boxes (address_id, description) VALUES
13
14     (5, 'Lagerbox vor der Schule'),
15     (6, 'Lagerbox neben dem Park'),
16     (7, 'Lagerbox bei der Bibliothek'),
17     (8, 'Lagerbox hinter dem Rathaus'),
18     (9, 'Lagerbox beim Museum'),
19     (10, 'Lagerbox an der Kirche'),
20     (11, 'Lagerbox am Marktplatz'),
21     (12, 'Lagerbox am Flussufer'),
22     (13, 'Lagerbox auf dem Hügel'),
23     (14, 'Lagerbox am Strand'),
24     (15, 'Lagerbox am Seeufer')
25
```

The screenshot shows a database interface with two main sections: a Result Grid and an Action Output log.

Result Grid: A table titled "tbl_boxes 1" with the following data:

box_id	address_id	description	deleted
1	5	Lagerbox vor der Schule	0
2	6	Lagerbox neben dem Park	0
3	7	Lagerbox bei der Bibliothek	0
4	8	Lagerbox hinter dem Rathaus	0
5	9	Lagerbox beim Museum	0
6	10	Lagerbox an der Kirche	0
7	11	Lagerbox am Marktplatz	0
8	12	Lagerbox am Flussufer	0
9	13	Lagerbox auf dem Hügel	0
10	14	Lagerbox am Strand	0
11	15	Lagerbox am Seeufer	0

Action Output: Log of SQL operations:

- # 1 16:42:41 CREATE TABLE dbo.tbl_boxes (box_id INT AUTO_INCREMENT PRIMARY KEY, address_id INT, description VARCHAR(255), ... 0 row(s) affected
- # 2 16:42:41 INSERT INTO dbo.tbl_boxes (address_id,description) VALUES (5,'Lagerbox vor der Schule'),(6,'Lagerbox neben dem Park'),(7,'Lagerbox bei der Bibliothek'),(8,'Lagerbox hinter dem Rathaus'),(9,'Lagerbox beim Museum'),(10,'Lagerbox an der Kirche'),(11,'Lagerbox am Marktplatz'),(12,'Lagerbox am Flussufer'),(13,'Lagerbox auf dem Hügel'),(14,'Lagerbox am Strand'),(15,'Lagerbox am Seeufer') 11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0
- # 3 16:42:45 SELECT * FROM dbo.tbl_boxes LIMIT 0,1000 11 row(s) returned

Abbildung 2: Struktur der Tabelle `tbl_boxes`

Tabelle: `tbl_user`

Die Tabelle `tbl_user` speichert Informationen über die Benutzer, die das Bibliothekssystem verwenden. Diese Tabelle enthält wichtige personenbezogene Daten und Verweise auf Adressen. Sie enthält die folgenden Felder:

- `user_id` - Eindeutige ID des Benutzers (Primärschlüssel)
- `first_name` - Vorname des Benutzers
- `last_name` - Nachname des Benutzers
- `email` - E-Mail-Adresse des Benutzers
- `phone_number` - Telefonnummer des Benutzers
- `address_id` - Verknüpfte Adresse des Benutzers (Fremdschlüssel)
- `birth_date` - Geburtsdatum des Benutzers
- `deleted` - Löschkennzeichen, um zu markieren, ob ein Benutzerkonto gelöscht wurde

```

1
2  -- Table for storing user information
3  ◎ CREATE TABLE dbo.tbl_user (
4      user_id INT AUTO_INCREMENT PRIMARY KEY,
5      first_name VARCHAR(50),
6      last_name VARCHAR(50),
7      email VARCHAR(100),
8      phone_number VARCHAR(50),
9      address_id INT,
10     birth_date DATE,
11     deleted BOOLEAN DEFAULT FALSE,
12     CONSTRAINT fk_user_address FOREIGN KEY (address_id) REFERENCES dbo.tbl_addresses(address_id)
13 );
14  -- Insert data into dbo.tbl_user
15  ◎ INSERT INTO dbo.tbl_user (first_name, last_name, email, phone_number, address_id, birth_date) VALUES
16      ('Max', 'Mustermann', 'max.mustermann@example.com', '0664-1234567', 1, '1990-01-01'),
17      ('Erika', 'Mustermann', 'erika.mustermann@example.com', '0664-2345678', 1, '1985-02-02'),
18      ('Hans', 'Müller', 'hans.mueller@example.com', '0664-3456789', 2, '1995-03-03'),
19      ('Anna', 'Müller', 'anna.mueller@example.com', '0664-6789012', 2, '1991-06-06'),
20      ('Petra', 'Schmidt', 'petra.schmidt@example.com', '0664-4567890', 3, '1980-04-04'),
21      ('Sebastian', 'Schmidt', 'sebastian.schmidt@example.com', '0664-9812345', 3, '1988-09-09'),
22      ('Thomas', 'Bauer', 'thomas.bauer@example.com', '0664-5678901', 4, '1992-05-05'),
23      ('Markus', 'Bauer', 'markus.bauer@example.com', '0664-7890123', 4, '1987-07-07'),
24      ('Julia', 'Weber', 'julia.weber@example.com', '0664-8901234', 5, '1993-08-08'),
25      ('Laura', 'Weber', 'laura.weber@example.com', '0664-0123456', 5, '1994-10-10');
26

```

user_id	first_name	last_name	email	phone_number	address_id	birth_date	deleted
1	Max	Mustermann	max.mustermann@example.com	0664-1234567	1	1990-01-01	0
2	Erika	Mustermann	erika.mustermann@example.com	0664-2345678	1	1985-02-02	0
3	Hans	Müller	hans.mueller@example.com	0664-3456789	2	1995-03-03	0
4	Anna	Müller	anna.mueller@example.com	0664-6789012	2	1991-06-06	0
5	Petra	Schmidt	petra.schmidt@example.com	0664-4567890	3	1980-04-04	0
6	Sebastian	Schmidt	sebastian.schmidt@example.com	0664-9812345	3	1988-09-09	0
7	Thomas	Bauer	thomas.bauer@example.com	0664-5678901	4	1992-05-05	0
8	Markus	Bauer	markus.bauer@example.com	0664-7890123	4	1987-07-07	0
9	Julia	Weber	julia.weber@example.com	0664-8901234	5	1993-08-08	0
10	Laura	Weber	laura.weber@example.com	0664-0123456	5	1994-10-10	0

tbl_user 1 ×

Output

#	Time	Action	Message
1	16:44:44	CREATE TABLE dbo.tbl_user (user_id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), phone_number VARCHAR(50), address_id INT, birth_date DATE, deleted BOOLEAN)	0 row(s) affected
2	16:44:44	INSERT INTO dbo.tbl_user (first_name, last_name, email, phone_number, address_id, birth_date) VALUES ('Max', 'Mustermann', 'max.mustermann@example.com', '0664-1234567', 1, '1990-01-01')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
3	16:44:50	SELECT * FROM dbo.tbl_user LIMIT 0,1000	10 row(s) returned

Abbildung 3: Struktur der Tabelle `tbl_user`

Tabelle: tbl_categories

Die Tabelle `tbl_categories` speichert die verschiedenen Kategorien der Bücher. Diese Tabelle hilft dabei, die Bücher in verschiedene Genres und Themenbereiche zu unterteilen. Sie enthält die folgenden Felder:

- `category_id` - Eindeutige ID der Kategorie (Primärschlüssel)
- `name` - Kategorienname
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Kategorie gelöscht wurde

```
1  -- Table for storing book categories
2  CREATE TABLE dbo.tbl_categories (
3      category_id INT AUTO_INCREMENT PRIMARY KEY,
4      name VARCHAR(50),
5      deleted BOOLEAN DEFAULT FALSE
6  );
7
8  -- Insert data into tbl_categories
9  INSERT INTO dbo.tbl_categories (name) VALUES
10 ('Krimi'),
11 ('Fantasy'),
12 ('Science-Fiction'),
13 ('Kochen'),
14 ('Reise'),
15 ('Kunst'),
16 ('Psychologie'),
17 ('Philosophie'),
18 ('Thriller'),
19 ('Politik');
20
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

category_id	name	deleted
1	Krimi	0
2	Fantasy	0
3	Science-Fiction	0
4	Kochen	0
5	Reise	0
6	Kunst	0
7	Psychologie	0
8	Philosophie	0
9	Thriller	0
10	Politik	0
NULL	NULL	NULL

!_categories 1 x

Output

#	Time	Action	Message
1	16:46:12	CREATE TABLE dbo.tbl_categories (category_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50), deleted BOOLEAN)	0 row(s) affected
2	16:46:12	INSERT INTO dbo.tbl_categories (name) VALUES ('Krimi'), ('Fantasy'), ('Science-Fiction'), ('Kochen'), ('Reise'), ('Kunst'), ('Psychologie'), ('Philosophie'), ('Thriller'), ('Politik')	10 rows(s) affected Records: 10 Duplicates: 0 Warnings: 0
3	16:46:17	SELECT * FROM dbo.tbl_categories LIMIT 0, 1000	10 row(s) returned

Abbildung 4: Struktur der Tabelle `tbl_categories`

Tabelle: tbl_languages

Die Tabelle `tbl_languages` speichert die verschiedenen Sprachen, in denen die Bücher verfügbar sind. Diese Tabelle ermöglicht die Zuordnung von Büchern zu ihrer jeweiligen Sprache. Sie enthält die folgenden Felder:

- `language_id` - Eindeutige ID der Sprache (Primärschlüssel)
- `language_code` - ISO-Code der Sprache
- `language_name` - Name der Sprache
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Sprache gelöscht wurde

```
1  -- Table for storing languages
2  CREATE TABLE dbo.tbl_languages (
3      language_id INT AUTO_INCREMENT PRIMARY KEY,
4      language_code CHAR(2),
5      language_name VARCHAR(50),
6      deleted BOOLEAN DEFAULT FALSE
7  );
8
9  -- Insert data into tbl_languages
10 INSERT INTO dbo.tbl_languages (language_code, language_name) VALUES
11  ('DE', 'Deutsch'),
12  ('EN', 'Englisch'),
13  ('FR', 'Französisch'),
14  ('ES', 'Spanisch'),
15  ('IT', 'Italienisch'),
16  ('RU', 'Russisch'),
17  ('PT', 'Portugiesisch'),
18  ('NL', 'Niederländisch'),
19  ('SV', 'Schwedisch'),
20  ('PL', 'Polnisch')
21
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window containing the SQL script for creating the `tbl_languages` table and inserting data. Below it is a data grid titled "languages" showing the 10 inserted rows. The data grid has columns: `language_id`, `language_code`, `language_name`, and `deleted`. The last row is a summary row. At the bottom, there is a "Logs" tab showing the execution history of the SQL statements.

language_id	language_code	language_name	deleted
1	DE	Deutsch	0
2	EN	Englisch	0
3	FR	Französisch	0
4	ES	Spanisch	0
5	IT	Italienisch	0
6	RU	Russisch	0
7	PT	Portugiesisch	0
8	NL	Niederländisch	0
9	SV	Schwedisch	0
10	PL	Polnisch	0

Logs

#	Time	Action	Message
1	17:57:57	CREATE TABLE dbo.tbl_languages (language_id INT AUTO_INCREMENT PRIMARY KEY, language_code CHAR(2), language_name VARCHAR(50), deleted BOOLEAN DEFAULT FALSE)	0 row(s) affected
2	17:57:57	INSERT INTO dbo.tbl_languages (language_code, language_name) VALUES ('DE', 'Deutsch'), ('EN', 'Englisch'), ('FR', 'Französisch'), ('ES', 'Spanisch'), ('IT', 'Italienisch'), ('RU', 'Russisch'), ('PT', 'Portugiesisch'), ('NL', 'Niederländisch'), ('SV', 'Schwedisch'), ('PL', 'Polnisch')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
3	17:58:07	SELECT * FROM dbo.tbl_languages LIMIT 0, 1000	10 row(s) returned

Abbildung 5: Struktur der Tabelle `tbl_languages`

Tabelle: tbl_authors

Die Tabelle `tbl_authors` speichert Informationen über die Autoren der Bücher. Diese Tabelle enthält wichtige Informationen über die Herkunft und die Sprache der Autoren. Sie enthält die folgenden Felder:

- `author_id` - Eindeutige ID des Autors (Primärschlüssel)
- `first_name` - Vorname des Autors
- `last_name` - Nachname des Autors
- `origin` - Herkunft des Autors
- `language_id` - Verknüpfte Sprache des Autors (Fremdschlüssel)
- `deleted` - Löschkennzeichen, um zu markieren, ob ein Autor gelöscht wurde

```
1  -- Table for storing authors
2  CREATE TABLE dbo.tbl_authors (
3      author_id INT AUTO_INCREMENT PRIMARY KEY,
4      first_name VARCHAR(50),
5      last_name VARCHAR(50),
6      origin VARCHAR(50),
7      language_id INT,
8      deleted BOOLEAN DEFAULT FALSE,
9      FOREIGN KEY (language_id) REFERENCES dbo.tbl_languages(language_id)
10 );
11
12 -- Insert data into tbl_authors
13 INSERT INTO tbl_authors (first_name, last_name, origin, language_id) VALUES
14
15 ('Franz', 'Kafka', 'Austria', 1),
16 ('Johann Wolfgang', 'von Goethe', 'Germany', 1),
17 ('Friedrich', 'Schiller', 'Germany', 1),
18 ('Hermann', 'Hesse', 'Germany', 1),
19 ('Thomas', 'Mann', 'Germany', 1),
20 ('Leo', 'Tolstoy', 'Russia', 1),
21 ('Fyodor', 'Dostoevsky', 'Russia', 1),
22 ('William', 'Shakespeare', 'England', 2),
23 ('Jane', 'Austen', 'England', 2),
24 ('Charles', 'Dickens', 'England', 2);
25
```

The screenshot shows a database interface with three main sections: SQL code, a result grid, and an output log.

- SQL Code:** The top section displays the creation script for the `tbl_authors` table, including its primary key, columns, and data insertion logic.
- Result Grid:** The middle section shows a table with 10 rows of data, each representing an author. The columns are `author_id`, `first_name`, `last_name`, `origin`, `language_id`, and `deleted`. The data includes Franz Kafka (Austria), Johann Wolfgang von Goethe (Germany), Friedrich Schiller (Germany), Hermann Hesse (Germany), Thomas Mann (Germany), Leo Tolstoy (Russia), Fyodor Dostoevsky (Russia), William Shakespeare (England), Jane Austen (England), and Charles Dickens (England).
- Output Log:** The bottom section displays the execution log with three entries:
 - CREATE TABLE (0 row(s) affected)
 - INSERT INTO (10 row(s) affected, 10 Duplicates: 0 Warnings: 0)
 - SELECT * (10 row(s) returned)

Abbildung 6: Struktur der Tabelle `tbl_authors`

Tabelle: tbl_book

Die Tabelle `tbl_book` speichert Informationen über die Bücher. Diese Tabelle enthält umfangreiche Informationen wie Titel, Veröffentlichungsjahr, ISBN, Kategorie, Sprache, Besitzer, Autor und Standort. Sie enthält die folgenden Felder:

- `book_id` - Eindeutige ID des Buches (Primärschlüssel)
- `title` - Buchtitel
- `publication_year` - Veröffentlichungsjahr
- `isbn` - Internationale Standardbuchnummer
- `category_id` - Verknüpfte Kategorie (Fremdschlüssel)
- `read_status` - Lesestatus (true = am Lesen, false = nicht am Lesen)
- `deleted` - Löschkennzeichen, um zu markieren, ob ein Buch gelöscht wurde
- `language_id` - Verknüpfte Sprache (Fremdschlüssel)
- `owner_id` - Besitzer des Buches (Fremdschlüssel)
- `author_id` - Verknüpfter Autor (Fremdschlüssel)
- `address_id` - Standort des Buches (Fremdschlüssel)

```

1  -- Table for storing book information
2  • CREATE TABLE dbo.tbl_book (
3      book_id INT AUTO_INCREMENT PRIMARY KEY,
4      title VARCHAR(100),
5      publication_year INT,
6      isbn varchar(13),
7      category_id INT,
8      read_status BOOLEAN DEFAULT FALSE,
9      deleted BOOLEAN DEFAULT FALSE,
10     language_id INT,
11     owner_id INT,
12     author_id INT,
13     address_id INT,
14     FOREIGN KEY (language_id) REFERENCES dbo.tbl_languages(language_id),
15     FOREIGN KEY (category_id) REFERENCES dbo.tbl_categories(category_id),
16     FOREIGN KEY (owner_id) REFERENCES dbo.tbl_user(user_id),
17     FOREIGN KEY (author_id) REFERENCES dbo.tbl_authors(author_id),
18     FOREIGN KEY (address_id) REFERENCES dbo.tbl_addresses(address_id)
19  );
20
21  -- Insert data into tbl_book
22  • INSERT INTO dbo.tbl_book (title, publication_year, isbn, category_id, language_id, owner_id, author_id, address_id)
23  ('The Trial', 1925, '9783150000113', 9, 1, 1, 1, 1),
24  ('The Metamorphosis', 1915, '9783150000120', 2, 1, 1, 1, 2),
25  ('Faust', 1808, '9783150000137', 8, 1, 1, 2, 3),
26  ('Die Leiden des jungen Werthers', 1774, '9783150000144', 1, 1, 1, 2, 4),
27  ('Die Räuber', 1781, '9783150000151', 9, 1, 1, 3, 5),
28  ('Siddhartha', 1922, '9783150000168', 7, 1, 1, 4, 6),
29  ('Der Steppenwolf', 1927, '9783150000175', 8, 1, 1, 4, 7),
30  ('Der Zauberberg', 1924, '9783150000182', 3, 1, 1, 5, 8),
31  ('Buddenbrooks', 1901, '9783150000199', 10, 1, 1, 5, 9),
32  ('War and Peace', 1869, '9783150000205', 5, 1, 1, 6, 10);
33

```

The screenshot shows the MySQL Workbench interface with two main sections: 'Result Grid' and 'Action Output'.

Result Grid: Displays the data from the `tbl_book` table. The columns are: book_id, title, publication_year, isbn, category_id, read_status, language_id, owner_id, author_id, and address_id. The data rows correspond to the entries listed in the SQL code above.

book_id	title	publication_year	isbn	category_id	read_status	language_id	owner_id	author_id	address_id
1	The Trial	1925	9783150000113	9	0	0	1	1	1
2	The Metamorphosis	1915	9783150000120	2	0	0	1	1	2
3	Faust	1808	9783150000137	8	0	0	1	1	2
4	Die Leiden des jungen Werthers	1774	9783150000144	1	0	0	1	1	3
5	Die Räuber	1781	9783150000151	9	0	0	1	1	4
6	Siddhartha	1922	9783150000168	7	0	0	1	1	5
7	Der Steppenwolf	1927	9783150000175	8	0	0	1	1	6
8	Der Zauberberg	1924	9783150000182	3	0	0	1	1	7
9	Buddenbrooks	1901	9783150000199	10	0	0	1	1	8
10	War and Peace	1869	9783150000205	5	0	0	1	1	9

Action Output: Shows the log of actions performed on the database:

- 1 16:55:32 CREATE TABLE dbo.tbl_book (book_id INT AUTO_INCREMENT PRIMARY KEY, title VARCHAR(100), publication_year INT, ... 0 row(s) affected
- 2 16:55:32 INSERT INTO dbo.tbl_book (title, publication_year, isbn, category_id, language_id, owner_id, author_id, address_id) VALUES ('The Trial', ..., 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
- 3 16:55:36 SELECT * FROM dbo.tbl_book LIMIT 0,1000 10 row(s) returned

Abbildung 7: Struktur der Tabelle `tbl_book`

Tabelle: tbl_borrowings

Die Tabelle `tbl_borrowings` speichert Informationen über die Ausleihen der Bücher. Diese Tabelle enthält wichtige Informationen darüber, wann und von wem ein Buch ausgeliehen wurde. Sie enthält die folgenden Felder:

- `borrowing_id` - Eindeutige ID der Ausleihe (Primärschlüssel)
- `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
- `book_id` - Verweis auf das ausgeliehene Buch (Fremdschlüssel)
- `borrow_date` - Datum der Ausleihe
- `return_date` - Datum der Rückgabe
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Ausleihvorgang gelöscht wurde

```
1  -- Table for storing book borrowing records
2  • CREATE TABLE dbo.tbl_borrowings (
3      borrowing_id INT AUTO_INCREMENT PRIMARY KEY,
4      user_id INT,
5      book_id INT,
6      borrow_date DATE,
7      return_date DATE,
8      deleted BOOLEAN DEFAULT FALSE,
9      FOREIGN KEY (user_id) REFERENCES dbo.tbl_user(user_id),
10     FOREIGN KEY (book_id) REFERENCES dbo.tbl_book(book_id)
11 );
12  -- Insert data into tbl_borrowings
13 •   INSERT INTO dbo.tbl_borrowings (user_id, book_id, borrow_date, return_date) VALUES
14     (1, 6, '2024-06-01', '2024-06-06'),
15     (2, 4, '2024-06-02', '2024-06-12'),
16     (3, 3, '2024-06-03', '2024-06-07'),
17     (4, 2, '2024-06-04', '2024-06-19'),
18     (5, 5, '2024-06-05', '2024-06-15'),
19     (1, 6, '2024-06-06', '2024-06-20'),
20     (2, 7, '2024-06-07', '2024-06-17'),
21     (3, 8, '2024-06-08', '2024-06-18'),
22     (4, 9, '2024-06-09', '2024-06-14'),
23     (5, 10, '2024-06-10', '2024-06-20')
24
```

borrowing_id	user_id	book_id	borrow_date	return_date	deleted
1	1	6	2024-06-01	2024-06-06	0
2	2	4	2024-06-02	2024-06-12	0
3	3	3	2024-06-03	2024-06-07	0
4	4	2	2024-06-04	2024-06-19	0
5	5	5	2024-06-05	2024-06-15	0
6	1	6	2024-06-06	2024-06-20	0
7	2	7	2024-06-07	2024-06-17	0
8	3	8	2024-06-08	2024-06-18	0
9	4	9	2024-06-09	2024-06-14	0
10	5	10	2024-06-10	2024-06-20	0

borrowings1 x
Input
Action Output
Time Action
1 16:57:19 CREATE TABLE dbo.tbl_borrowings (borrowing_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book_id INT, borrow_... 0 row(s) affected
2 16:57:19 INSERT INTO dbo.tbl_borrowings (user_id, book_id, borrow_date, return_date) VALUES (1, 6, '2024-06-01', '2024-06-06'), (2, 4, '2024-06-02...', 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
3 16:57:29 SELECT * FROM dbo.tbl_borrowings LIMIT 0, 1000 10 row(s) returned

Abbildung 8: Struktur der Tabelle `tbl_borrowings`

Tabelle: tbl_reservations

Die Tabelle `tbl_reservations` speichert Informationen über die Reservierungen der Bücher. Diese Tabelle enthält wichtige Informationen darüber, wann und von wem ein Buch reserviert wurde. Sie enthält die folgenden Felder:

- `reservation_id` - Eindeutige ID der Reservierung (Primärschlüssel)
 - `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
 - `book_id` - Verweis auf das reservierte Buch (Fremdschlüssel)
 - `reservation_date` - Datum der Reservierung
 - `status` - Status der Reservierung
 - `deleted` - Löschkennzeichen, um zu markieren, ob eine Reservierung gelöscht wurde

Abbildung 9: Struktur der Tabelle `tbl_reservations`

Tabelle: tbl_ratings

Die Tabelle `tbl_ratings` speichert Informationen über die Bewertungen der Bücher. Diese Tabelle enthält wichtige Informationen darüber, welche Bücher von welchem Benutzer bewertet wurden, inklusive der Bewertungen und Kommentare. Sie enthält die folgenden Felder:

- `rating_id` - Eindeutige ID der Bewertung (Primärschlüssel)
- `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
- `book_id` - Verweis auf das bewertete Buch (Fremdschlüssel)
- `rating` - Bewertungszahl (1-5 Sterne, 5 ist das Beste)
- `comment` - Kommentar zur Bewertung
- `rating_date` - Datum der Bewertungsabgabe
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Bewertung gelöscht wurde

```
1 -- Table for storing book ratings
2 * CREATE TABLE dbo.tbl_ratings (
3     rating_id INT AUTO_INCREMENT PRIMARY KEY,
4     user_id INT,
5     book_id INT,
6     rating INT,
7     comment TEXT,
8     rating_date DATE,
9     deleted BOOLEAN DEFAULT FALSE,
10    FOREIGN KEY (user_id) REFERENCES dbo.tbl_user(user_id),
11    FOREIGN KEY (book_id) REFERENCES dbo.tbl_book(book_id)
12 ) ;
13
14 -- Insert data into tbl_ratings
15 * INSERT INTO dbo.tbl_ratings (user_id, book_id, rating, comment, rating_date) VALUES
16   ('1, 2, 5, 'Ein Meisterwerk!', '2024-01-15'),
17   ('2, 6, 4, 'Spannende Handlung.', '2024-01-20'),
18   ('3, 7, 4, 'Sehr gut geschrieben.', '2024-02-15'),
19   ('4, 2, 3, 'Interessante Charaktere.', '2024-02-20'),
20   ('5, 3, 3, 'Gute Unterhaltung.', '2024-03-15'),
21   ('1, 3, 2, 'Etwas langatig.', '2024-03-20'),
22   ('2, 4, 8, 'Könnte mich nicht überzeugen.', '2024-04-15'),
23   ('3, 4, 7, 'Schwache Handlung.', '2024-04-20'),
24   ('4, 5, 6, 'Hat mir nicht gefallen.', '2024-05-15'),
25   ('5, 5, 3, 'Interessante Ideen.', '2024-05-20');
26
```

Result Grid | Filter Rows | Export/Imports | Wrap Cell Content | ID

rating_id	user_id	book_id	rating	comment	rating_date	deleted
1	1	2	5	Ein Meisterwerk!	2024-01-15	0
2	2	6	4	Spannende Handlung.	2024-01-20	0
3	3	7	4	Sehr gut geschrieben.	2024-02-15	0
4	4	2	3	Interessante Charaktere.	2024-02-20	0
5	5	3	3	Gute Unterhaltung.	2024-03-15	0
6	1	3	2	Etwas langatig.	2024-03-20	0
7	2	4	8	Könnte mich nicht überzeugen.	2024-04-15	0
8	3	4	7	Schwache Handlung.	2024-04-20	0
9	4	5	6	Hat mir nicht gefallen.	2024-05-15	0
10	5	5	3	Interessante Ideen.	2024-05-20	0

tbl_ratings 1 <--> Output

Action Output | Time Action

1 18:06:13 CREATE TABLE dbo.tbl_ratings (rating_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book_id INT, rating INT, comment TEXT, rating_date DATE, deleted BOOLEAN DEFAULT FALSE) 0 rows affected

2 18:06:13 INSERT INTO dbo.tbl_ratings (user_id, book_id, rating, comment, rating_date) VALUES (1, 2, 5, 'Ein Meisterwerk!', '2024-01-15'), (2, 6, 4, 'Spannende Handlung.', '2024-01-20), (3, 7, 4, 'Sehr gut geschrieben.', 2024-02-15), (4, 2, 3, 'Interessante Charaktere.', '2024-02-20'), (5, 3, 3, 'Gute Unterhaltung.', '2024-03-15'), (1, 3, 2, 'Etwas langatig.', '2024-03-20'), (2, 4, 8, 'Könnte mich nicht überzeugen.', '2024-04-15'), (3, 4, 7, 'Schwache Handlung.', '2024-04-20'), (4, 5, 6, 'Hat mir nicht gefallen.', '2024-05-15'), (5, 5, 3, 'Interessante Ideen.', '2024-05-20') 10 rows affected 10 row(s) returned

Abbildung 10: Struktur der Tabelle `tbl_ratings`

Tabelle: tbl_position_log

Die Tabelle `tbl_position_log` speichert Informationen über die Änderungen der Positionen der Bücher. Diese Tabelle ermöglicht es, den Weg eines Buches nachzuverfolgen, indem es die vorherige und die neue Position aufzeichnet. Sie enthält die folgenden Felder:

- `position_log_id` - Eindeutige ID des Logeintrags (Primärschlüssel)
- `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
- `book_id` - Verweis auf das Buch (Fremdschlüssel)
- `log_date` - Datum der Positionsänderung
- `previous_address_id` - Vorherige Position des Buches (Fremdschlüssel)
- `new_address_id` - Neue Position des Buches (Fremdschlüssel)

```
1  ◻ CREATE TABLE dbo.tbl_position_log (
2      position_log_id INT AUTO_INCREMENT PRIMARY KEY,
3      user_id INT,
4      book_id INT,
5      log_date DATE,
6      previous_address_id INT,
7      new_address_id INT,
8      FOREIGN KEY (user_id) REFERENCES dbo.tbl_user(user_id),
9      FOREIGN KEY (book_id) REFERENCES dbo.tbl_book(book_id),
10     FOREIGN KEY (previous_address_id) REFERENCES dbo.tbl_addresses(address_id),
11     FOREIGN KEY (new_address_id) REFERENCES dbo.tbl_addresses(address_id)
12   );
13
14  -- Insert data into tbl_position_log
15  •  INSERT INTO dbo.tbl_position_log (user_id, book_id, log_date, previous_address_id, new_address_id) VALUES
16    (1, 1, '2024-06-01', 1, 2),
17    (2, 2, '2024-06-03', 2, 3),
18    (3, 3, '2024-06-05', 3, 4),
19    (4, 4, '2024-06-07', 4, 5),
20    (5, 5, '2024-06-09', 5, 6),
21    (1, 1, '2024-06-11', 2, 3),
22    (2, 2, '2024-06-13', 3, 4),
23    (3, 3, '2024-06-15', 4, 5),
24    (4, 4, '2024-06-17', 5, 6),
25    (5, 5, '2024-06-19', 6, 7);
26
27
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | position_log_1 x

position_log_id	user_id	book_id	log_date	previous_address_id	new_address_id
1	1	1	2024-06-01	1	2
2	2	2	2024-06-03	2	3
3	3	3	2024-06-05	3	4
4	4	4	2024-06-07	4	5
5	5	5	2024-06-09	5	6
6	1	1	2024-06-11	2	3
7	2	2	2024-06-13	3	4
8	3	3	2024-06-15	4	5
9	4	4	2024-06-17	5	6
10	5	5	2024-06-19	6	7

Output

#	Time	Action	Message
1	17:03:46	CREATE TABLE dbo.tbl_position_log (position_log_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book_id INT, log_d...)	0 row(s) affected
2	17:03:46	INSERT INTO dbo.tbl_position_log (user_id, book_id, log_date, previous_address_id, new_address_id) VALUES (1, 1, 2024-06-01, 1, 2), (2, 2, 2024-06-03, 2, 3), (3, 3, 2024-06-05, 3, 4), (4, 4, 2024-06-07, 4, 5), (5, 5, 2024-06-09, 5, 6), (1, 1, 2024-06-11, 2, 3), (2, 2, 2024-06-13, 3, 4), (3, 3, 2024-06-15, 4, 5), (4, 4, 2024-06-17, 5, 6), (5, 5, 2024-06-19, 6, 7)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0
3	17:03:51	SELECT * FROM dbo.tbl_position_log LIMIT 0, 1000	10 row(s) returned

Abbildung 11: Struktur der Tabelle `tbl_position_log`

Tabelle: tbl_read_status_log

Die Tabelle `tbl_read_status_log` speichert Informationen über Änderungen im Lesestatus der Bücher. Diese Tabelle zeichnet die vorherigen und neuen Lesestatus der Bücher auf, um Änderungen nachverfolgen zu können. Sie enthält die folgenden Felder:

- `read_status_log_id` - Eindeutige ID des Logeintrags (Primärschlüssel)
- `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
- `book_id` - Verweis auf das Buch (Fremdschlüssel)
- `previous_read_status` - Vorheriger Lesestatus
- `new_read_status` - Neuer Lesestatus
- `log_date` - Datum der Statusänderung

```
1  -- Table for logging changes in book read statuses
2 • CREATE TABLE dbo.tbl_read_status_log (
3     read_status_log_id INT AUTO_INCREMENT PRIMARY KEY,
4     user_id INT,
5     book_id INT,
6     previous_read_status BOOLEAN,
7     new_read_status BOOLEAN,
8     log_date DATE,
9     FOREIGN KEY (user_id) REFERENCES dbo.tbl_user(user_id),
10    FOREIGN KEY (book_id) REFERENCES dbo.tbl_book(book_id)
11  )
12  -- Insert data into tbl_read_status_log
13 • INSERT INTO dbo.tbl_read_status_log (user_id, book_id, previous_read_status, new_read_status, log_date) VALUES
14  (1, 1, FALSE, TRUE, '2024-06-02'),
15  (2, 2, FALSE, TRUE, '2024-06-04'),
16  (3, 3, FALSE, TRUE, '2024-06-06'),
17  (4, 4, FALSE, TRUE, '2024-06-08'),
18  (5, 5, FALSE, TRUE, '2024-06-10'),
19  (1, 1, TRUE, FALSE, '2024-06-12'),
20  (2, 2, TRUE, FALSE, '2024-06-14'),
21  (3, 3, TRUE, FALSE, '2024-06-16'),
22  (4, 4, TRUE, FALSE, '2024-06-18'),
23  (5, 5, TRUE, FALSE, '2024-06-20');
24
```

The screenshot shows the creation of the `tbl_read_status_log` table and its initial data insertion. The table has columns: `read_status_log_id`, `user_id`, `book_id`, `previous_read_status`, `new_read_status`, and `log_date`. The data consists of 10 rows, each representing a status change for a specific user and book.

read_status_log_id	user_id	book_id	previous_read_status	new_read_status	log_date
1	1	1	0	1	2024-06-02
2	2	2	0	1	2024-06-04
3	3	3	0	1	2024-06-06
4	4	4	0	1	2024-06-08
5	5	5	0	1	2024-06-10
6	1	1	1	0	2024-06-12
7	2	2	1	0	2024-06-14
8	3	3	1	0	2024-06-16
9	4	4	1	0	2024-06-18
10	5	5	1	0	2024-06-20

The output window shows the SQL commands used to create the table and insert the data, along with the results of the `SELECT * FROM dbo.tbl_read_status_log LIMIT 0, 1000` query.

#	Time	Action	Message
1	17:04:49	CREATE TABLE dbo.tbl_read_status_log (read_status_log_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book_id INT, ...)	0 row(s) affected
2	17:04:49	INSERT INTO dbo.tbl_read_status_log (user_id, book_id, previous_read_status, new_read_status, log_date) VALUES (1, 1, FALSE, TRUE, '2024-06-02')	10 row(s) affected Records: 10 Warnings: 0
3	17:04:53	SELECT * FROM dbo.tbl_read_status_log LIMIT 0, 1000	10 row(s) returned

Abbildung 12: Struktur der Tabelle `tbl_read_status_log`

Tabelle: **tbl_wishlists**

Die Tabelle `tbl_wishlists` speichert Informationen über die Wunschlisten der Benutzer. Diese Tabelle ermöglicht es den Benutzern, Bücher zu markieren, die sie gerne lesen möchten. Sie enthält die folgenden Felder:

- `wishlist_id` - Eindeutige ID des Wunscheintrags (Primärschlüssel)
 - `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
 - `book_id` - Verweis auf das gewünschte Buch (Fremdschlüssel)
 - `added` - Datum, an dem das Buch zur Wunschliste hinzugefügt wurde
 - `deleted` - Löschkennzeichen, um zu markieren, ob ein Wunscheintrag gelöscht wurde

```
1 -- Table for storing user wishlists
2 • CREATE TABLE dbo.tbl_wishlists (
3     wishlist_id INT AUTO_INCREMENT PRIMARY KEY,
4     user_id INT,
5     book_id INT,
6     added DATE,
7     deleted BOOLEAN DEFAULT FALSE,
8     FOREIGN KEY (user_id) REFERENCES dbo.tbl_user(user_id),
9     FOREIGN KEY (book_id) REFERENCES dbo.tbl_book(book_id)
10 )IS
11
12 -- Insert data into tbl_wishlists
13 • INSERT INTO dbo.tbl_wishlists (user_id, book_id, added) VALUES
14 (1, 1, '2024-06-01'),
15 (2, 1, '2024-06-02'),
16 (3, 1, '2024-06-03'),
17 (4, 1, '2024-06-24'),
18 (5, 1, '2024-06-04'),
19 (6, 1, '2024-06-05'),
20 (3, 1, '2024-06-06'),
21 (3, 1, '2024-06-07'),
22 (4, 1, '2024-06-08'),
23 (1, 1, '2024-06-10')
```

Result Grid Filter Rows Export/Import Wrap Cell Content

wishlist_id	user_id	book_id	added	deleted
1	1	1	2024-06-01	0
2	1	1	2024-06-02	0
3	1	1	2024-06-03	0
4	4	1	2024-06-24	0
5	5	1	2024-06-01	0
6	1	1	2024-06-05	0
7	2	1	2024-06-04	0
8	3	1	2024-06-08	0
9	4	1	2024-06-09	0
10	5	1	2024-06-10	0

Sql wishlist1 x Output

Action Output Time Action

1 18:20:57 CREATE TABLE dbo.wishlists (wishlist_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book_id INT, added DATE, deleted BOOLEAN DEFAULT FALSE, FOREIGN KEY (user_id) REFERENCES dbo.user(user_id), FOREIGN KEY (book_id) REFERENCES dbo.book(book_id))IS

2 18:20:57 INSERT INTO dbo.wishlists (user_id, book_id, added) VALUES (1, 1, '2024-06-01'), (2, 1, '2024-06-02'), (3, 1, '2024-06-03'), (4, 1, '2024-06-24'), (5, 1, '2024-06-04'), (6, 1, '2024-06-05'), (3, 1, '2024-06-06'), (3, 1, '2024-06-07'), (4, 1, '2024-06-08'), (1, 1, '2024-06-10')

3 18:21:01 SELECT * FROM dbo.wishlists LIMIT 0,1000

Message

1 row(s) affected

10 rows affected Records: 10 Duplicates: 0 Warnings: 0

10 rows returned

10 rows returned

Abbildung 13: Struktur der Tabelle `tbl_wishlists`

Tabelle: tbl_recommendations

Die Tabelle `tbl_recommendations` speichert Empfehlungen für Bücher. Diese Tabelle ermöglicht es den Benutzern, Bücher basierend auf ihren Vorlieben zu empfehlen. Sie enthält die folgenden Felder:

- `recommendations_id` - Eindeutige ID der Empfehlung (Primärschlüssel)
- `user_id` - Verweis auf den Benutzer (Fremdschlüssel)
- `book_id` - Verweis auf das gelesene Buch (Fremdschlüssel)
- `recommendation_book_id` - Verweis auf das empfohlene Buch (Fremdschlüssel)
- `deleted` - Löschkennzeichen, um zu markieren, ob eine Empfehlung gelöscht wurde

The screenshot shows the SQL Server Management Studio interface. At the top, there is a code editor window containing the following SQL script:

```
1 -- Table for storing book recommendations
2
3 • CREATE TABLE dbo.tbl_recommendations (
4     recommendations_id INT AUTO_INCREMENT PRIMARY KEY,
5     user_id INT,
6     book_id INT,
7     recommendation_book_id INT,
8     deleted BOOLEAN DEFAULT FALSE,
9     FOREIGN KEY (user_id) REFERENCES dbo.tbl_user(user_id),
10    FOREIGN KEY (book_id) REFERENCES dbo.tbl_book(book_id),
11    FOREIGN KEY (recommendation_book_id) REFERENCES dbo.tbl_book(book_id)
12 );
13
14 -- Insert data into tbl_recommendations
15
16 • INSERT INTO dbo.tbl_recommendations (user_id, book_id, recommendation_book_id) VALUES
17     (1, 1, 1),
18     (2, 2, 2),
19     (3, 3, 3),
20     (4, 4, 4),
21     (5, 5, 5),
22     (1, 2, 2),
23     (2, 3, 3),
24     (3, 4, 4),
25     (4, 5, 5),
26     (5, 6, 6)
```

Below the script is a results grid showing the inserted data:

recommendations_id	user_id	book_id	recommendation_book_id	deleted
1	1	1	1	0
2	2	2	2	0
3	3	3	3	0
4	4	4	4	0
5	5	5	5	0
6	1	6	6	0
7	2	7	7	0
8	3	8	8	0
9	4	9	9	0
10	5	10	10	0
*	XXXX	XXXX	XXXX	XXXX

At the bottom, there is an output window showing the execution history:

Time	Action	Message
14:37:34	CREATE TABLE dbo.tbl_recommendations (recommendations_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book_id INT, recommendation_book_id INT, deleted BOOLEAN DEFAULT ...)	0 row(s) affected
14:37:34	INSERT INTO dbo.tbl_recommendations (user_id, book_id, recommendation_book_id) VALUES (1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (5, 5, 5), (1, 6, 6), (2, 7, 7), (3, 8, 8), (4, 9, 9), (5, 10, 10)	10 rows affected Records: 10 Duplicates: 0 Warnings: 0
14:37:43	SELECT * FROM dbo.tbl_recommendations LIMIT 0, 1000	10 rows returned
14:38:03	SELECT * FROM dbo.tbl_recommendations LIMIT 0, 1000	10 rows returned

Abbildung 14: Struktur der Tabelle `tbl_recommendations`

Testfälle

Wie schon im Feedback erwähnt hätten wir durch die Überlegung der Testfälle in der Konzeptionsphase wichtige Erkenntnisse gewinnen können. Zum Beispiel stellte sich während eines Falles heraus dass die `tbl_wishlists` ohne einen Dummy-User der Besitzer dieser Bücher ist, nicht sinnvoll befüllt werden konnte. Das zeigt wie wichtig es ist solche Details schon frühzeitig zu berücksichtigen um spätere Anpassungen zu vermeiden. Im Rahmen unserer Tests haben wir erfolgreich verschiedene Funktionalitäten der Datenbank überprüft, darunter das Anlegen eines neuen Buches samt Autor, die Aktualisierung von Lese-Status-Logs, das Umlagern eines Buches von einer Wohnadresse in eine Lagerbox sowie das Hinzufügen von Büchern zur Wunschliste und das Reservieren von Büchern. Durch diese Tests konnten wir nahezu alle Tabellen der Datenbank abdecken. Die Tabellen `tbl_categories` und `tbl_boxes` wurden nicht separat getestet, da sie hauptsächlich als referenzierte Datenquellen für andere Tabellen dienen. Daher wurde der Fokus auf die direkten Interaktionen mit den Benutzer-, Buch-, und Log-Tabellen gelegt, um sicherzustellen, dass die wichtigsten Datenbankfunktionen zuverlässig funktionieren. Die Testfälle sind in einer SQL Datei angefügt.

Fazit

In dieser Phase stand die praktische Umsetzung der Spezifikation in eine funktionierende Datenbank im Mittelpunkt. Die Erarbeitung der SQL-Skripte verlief reibungslos, und mit MySQL konnten wir die Datenbank effektiv umsetzen. Dabei war es besonders wichtig, jede Anpassung konsequent und vollständig durchzuführen da Änderungen oft weitreichende Auswirkungen haben können. Ein Beispiel dafür war das Entfernen der Tabelle `tbl_locations`. Hier mussten alle Fremdschlüssele, die auf diese Tabelle verwiesen angepasst werden um das System funktionsfähig zu halten. Diese Phase verdeutlichte uns, wie entscheidend es ist, auch kleine Änderungen umfassend in allen betroffenen Bereichen der Datenbank umzusetzen. Zudem haben wir die Datentypen und Zeichenlänge angepasst um die Effizienz zu steigern. Während den Tests stellten wir fest das wir keine Bücher für die Wishlist anlegen können ohne ein Benutzer, wir entschieden uns daher für einen "Dummyuser" der diese Bücher besitzt.

In der letzten Phase wir das System dann abschließend nochmals mit dem Feedback überarbeitet. Das Portfolio finalisiert und in einem GitHub Repository bereitgestellt. Sowie die Technische Herangehensweise und den Ablauf der Arbeit Reflektiert.