

```

1  protected override void OnModelCreating(ModelBuilder modelBuilder)
2      {
3      modelBuilder.HasDefaultSchema("SBD_ST_PS5_1");
4
5      modelBuilder.Entity<Actor>(entity =>
6      {
7          entity.ToTable("ACTOR");
8
9          entity.Property(e => e.ActorId)
10             .HasColumnType("NUMBER")
11             .HasColumnName("ACTOR_ID");
12
13          entity.Property(e => e.Birthday)
14             .HasColumnType("DATE")
15             .HasColumnName("BIRTHDAY");
16
17          entity.Property(e => e.Firstname)
18             .IsRequired()
19             .HasMaxLength(50)
20             .HasColumnName("FIRSTNAME");
21
22          entity.Property(e => e.Lastname)
23             .IsRequired()
24             .HasMaxLength(50)
25             .HasColumnName("LASTNAME");
26
27          entity.Property(e => e.Nationality)
28             .HasMaxLength(40)
29             .HasColumnName("NATIONALITY");
30      });
31
32      modelBuilder.Entity<Actoraward>(entity =>
33      {
34          entity.HasKey(e => new { e.ActorActorId, e.AwardAwardId })
35             .HasName("ACTORAWARD_PK");
36
37          entity.ToTable("ACTORAWARD");
38
39          entity.Property(e => e.ActorActorId)
40             .HasColumnType("NUMBER")
41             .HasColumnName("ACTOR_ACTOR_ID");
42
43          entity.Property(e => e.AwardAwardId)
44             .HasColumnType("NUMBER")
45             .HasColumnName("AWARD_AWARD_ID");
46
47          entity.Property(e => e.Date).HasColumnType("DATE");
48
49          entity.HasOne(d => d.ActorActor)
50             .WithMany(p => p.Actorawards)
51             .HasForeignKey(d => d.ActorActorId)
52             .HasConstraintName("ACTORAWARD_ACTOR_FK");
53
54          entity.HasOne(d => d.AwardAward)
55             .WithMany(p => p.Actorawards)
56             .HasForeignKey(d => d.AwardAwardId)
57             .HasConstraintName("ACTORAWARD_AWARD_FK");
58      });
59
60      modelBuilder.Entity<Archive>(entity =>
61      {
62          entity.HasKey(e => new { e.UserUserId, e.ScreeningScreeningId })
63             .HasName("ARCHIVE_PK");
64
65          entity.ToTable("ARCHIVE");
66
67          entity.Property(e => e.UserUserId)
68             .HasColumnType("NUMBER")
69             .HasColumnName("USER_USER_ID");
70
71          entity.Property(e => e.ScreeningScreeningId)
72             .HasColumnType("NUMBER")
73             .HasColumnName("SCREENING_SCREENING_ID");

```

```

74         entity.Property(e => e.Date).HasColumnType("DATE");
75
76
77         entity.HasOne(d => d.ScreeningScreening)
78             .WithMany(p => p.Archives)
79             .HasForeignKey(d => d.ScreeningScreeningId)
80             .OnDelete(DeleteBehavior.ClientSetNull)
81             .HasConstraintName("ARCHIVE_SCREENING_FK");
82
83         entity.HasOne(d => d.UserUser)
84             .WithMany(p => p.Archives)
85             .HasForeignKey(d => d.UserUserId)
86             .OnDelete(DeleteBehavior.ClientSetNull)
87             .HasConstraintName("ARCHIVE_USER_FK");
88     });
89
90     modelBuilder.Entity<Article>(entity =>
91     {
92         entity.ToTable("ARTICLE");
93
94         entity.Property(e => e.ArticleId)
95             .HasColumnType("NUMBER")
96             .HasColumnName("ARTICLE_ID");
97
98         entity.Property(e => e.Content)
99             .IsRequired()
100             .HasMaxLength(1000)
101             .HasColumnName("CONTENT");
102
103         entity.Property(e => e.Date).HasColumnType("DATE");
104
105         entity.Property(e => e.WorkerWorkerId)
106             .HasColumnType("NUMBER")
107             .HasColumnName("WORKER_WORKER_ID");
108
109         entity.HasOne(d => d.WorkerWorker)
110             .WithMany(p => p.Articles)
111             .HasForeignKey(d => d.WorkerWorkerId)
112             .OnDelete(DeleteBehavior.ClientSetNull)
113             .HasConstraintName("ARTICLE_WORKER_FK");
114     });
115
116     modelBuilder.Entity<Award>(entity =>
117     {
118         entity.ToTable("AWARD");
119
120         entity.Property(e => e.AwardId)
121             .HasColumnType("NUMBER")
122             .HasColumnName("AWARD_ID");
123
124         entity.Property(e => e.Name)
125             .IsRequired()
126             .HasMaxLength(50)
127             .HasColumnName("NAME");
128     });
129
130     modelBuilder.Entity<Cinema>(entity =>
131     {
132         entity.ToTable("CINEMA");
133
134         entity.Property(e => e.CinemaId)
135             .HasColumnType("NUMBER")
136             .HasColumnName("CINEMA_ID");
137
138         entity.Property(e => e.City)
139             .IsRequired()
140             .HasMaxLength(50)
141             .HasColumnName("CITY");
142
143         entity.Property(e => e.Name)
144             .IsRequired()
145             .HasMaxLength(60)
146             .HasColumnName("NAME");

```

```

147     });
148
149     modelBuilder.Entity<Director>(entity =>
150     {
151         entity.ToTable("DIRECTOR");
152
153         entity.Property(e => e.DirectorId)
154             .HasColumnType("NUMBER")
155             .HasColumnName("DIRECTOR_ID");
156
157         entity.Property(e => e.Birthday)
158             .HasColumnType("DATE")
159             .HasColumnName("BIRTHDAY");
160
161         entity.Property(e => e.Firstname)
162             .IsRequired()
163             .HasMaxLength(50)
164             .HasColumnName("FIRSTNAME");
165
166         entity.Property(e => e.Lastname)
167             .IsRequired()
168             .HasMaxLength(50)
169             .HasColumnName("LASTNAME");
170
171         entity.Property(e => e.Nationality)
172             .HasMaxLength(40)
173             .HasColumnName("NATIONALITY");
174     });
175
176     modelBuilder.Entity<Directoraward>(entity =>
177     {
178         entity.HasKey(e => new { e.DirectorDirectorId, e.AwardAwardId })
179             .HasName("DIRECTORAWARD_PK");
180
181         entity.ToTable("DIRECTORAWARD");
182
183         entity.Property(e => e.DirectorDirectorId)
184             .HasColumnType("NUMBER")
185             .HasColumnName("DIRECTOR_DIRECTOR_ID");
186
187         entity.Property(e => e.AwardAwardId)
188             .HasColumnType("NUMBER")
189             .HasColumnName("AWARD_AWARD_ID");
190
191         entity.Property(e => e.Date).HasColumnType("DATE");
192
193         entity.HasOne(d => d.AwardAward)
194             .WithMany(p => p.Directorawards)
195             .HasForeignKey(d => d.AwardAwardId)
196             .HasConstraintName("DIRECTORAWARD_AWARD_FK");
197
198         entity.HasOne(d => d.DirectorDirector)
199             .WithMany(p => p.Directorawards)
200             .HasForeignKey(d => d.DirectorDirectorId)
201             .HasConstraintName("DIRECTORAWARD_DIRECTOR_FK");
202     });
203
204     modelBuilder.Entity<Genre>(entity =>
205     {
206         entity.ToTable("GENRE");
207
208         entity.Property(e => e.GenreId)
209             .HasColumnType("NUMBER")
210             .HasColumnName("GENRE_ID");
211
212         entity.Property(e => e.Name)
213             .IsRequired()
214             .HasMaxLength(50)
215             .HasColumnName("NAME");
216     });
217
218     modelBuilder.Entity<Languageversion>(entity =>
219     {

```

```

220         entity.HasKey(e => e.LvId)
221             .HasName("LANGUAGEVERSION_PK");
222
223         entity.ToTable("LANGUAGEVERSION");
224
225         entity.Property(e => e.LvId)
226             .HasColumnType("NUMBER")
227             .HasColumnName("LV_ID");
228
229         entity.Property(e => e.Language)
230             .IsRequired()
231             .HasMaxLength(50)
232             .HasColumnName("LANGUAGE");
233
234         entity.Property(e => e.Type)
235             .IsRequired()
236             .HasMaxLength(50)
237             .HasColumnName("TYPE");
238     });
239
240     modelBuilder.Entity<Movie>(entity =>
241     {
242         entity.ToTable("MOVIE");
243
244         entity.Property(e => e.MovieId)
245             .HasColumnType("NUMBER")
246             .HasColumnName("MOVIE_ID");
247
248         entity.Property(e => e.Country)
249             .HasMaxLength(35)
250             .HasColumnName("COUNTRY");
251
252         entity.Property(e => e.Description)
253             .HasMaxLength(1000)
254             .HasColumnName("DESCRIPTION");
255
256         entity.Property(e => e.GenreGenreId)
257             .HasColumnType("NUMBER")
258             .HasColumnName("GENRE_GENRE_ID");
259
260         entity.Property(e => e.Name)
261             .IsRequired()
262             .HasMaxLength(50)
263             .HasColumnName("NAME");
264
265         entity.Property(e => e.Productionyear)
266             .HasColumnType("DATE")
267             .HasColumnName("PRODUCTIONYEAR");
268
269         entity.HasOne(d => d.GenreGenre)
270             .WithMany(p => p.Movies)
271             .HasForeignKey(d => d.GenreGenreId)
272             .OnDelete(DeleteBehavior.ClientSetNull)
273             .HasConstraintName("MOVIE_GENRE_FK");
274     });
275
276     modelBuilder.Entity<Movieactor>(entity =>
277     {
278         entity.HasKey(e => new { e.MovieMovieId, e.ActorActorId })
279             .HasName("MOVIEACTOR_PK");
280
281         entity.ToTable("MOVIEACTOR");
282
283         entity.Property(e => e.MovieMovieId)
284             .HasColumnType("NUMBER")
285             .HasColumnName("MOVIE_MOVIE_ID");
286
287         entity.Property(e => e.ActorActorId)
288             .HasColumnType("NUMBER")
289             .HasColumnName("ACTOR_ACTOR_ID");
290
291         entity.Property(e => e.Role)
292             .IsRequired()

```

```

293         .HasMaxLength(50)
294         .HasColumnName("ROLE");
295
296     entity.HasOne(d => d.ActorActor)
297         .WithMany(p => p.Movieactors)
298         .HasForeignKey(d => d.ActorActorId)
299         .HasConstraintName("MOVIEACTOR_ACTOR_FK");
300
301     entity.HasOne(d => d.MovieMovie)
302         .WithMany(p => p.Movieactors)
303         .HasForeignKey(d => d.MovieMovieId)
304         .HasConstraintName("MOVIEACTOR_MOVIE_FK");
305 });
306
307 modelBuilder.Entity<Moviedirector>(entity =>
308 {
309     entity.HasKey(e => new { e.MovieMovieId, e.DirectorDirectorId })
310         .HasName("MOVIEDIRECTOR_PK");
311
312     entity.ToTable("MOVIEDIRECTOR");
313
314     entity.Property(e => e.MovieMovieId)
315         .HasColumnType("NUMBER")
316         .HasColumnName("MOVIE_MOVIE_ID");
317
318     entity.Property(e => e.DirectorDirectorId)
319         .HasColumnType("NUMBER")
320         .HasColumnName("DIRECTOR_DIRECTOR_ID");
321
322     entity.HasOne(d => d.DirectorDirector)
323         .WithMany(p => p.Moviedirectors)
324         .HasForeignKey(d => d.DirectorDirectorId)
325         .HasConstraintName("MOVIEDIRECTOR_DIRECTOR_FK");
326
327     entity.HasOne(d => d.MovieMovie)
328         .WithMany(p => p.Moviedirectors)
329         .HasForeignKey(d => d.MovieMovieId)
330         .HasConstraintName("MOVIEDIRECTOR_MOVIE_FK");
331 });
332
333 modelBuilder.Entity<Review>(entity =>
334 {
335     entity.ToTable("REVIEW");
336
337     entity.Property(e => e.ReviewId)
338         .HasColumnType("NUMBER")
339         .HasColumnName("REVIEW_ID");
340
341     entity.Property(e => e.Content)
342         .HasMaxLength(1000)
343         .HasColumnName("CONTENT");
344
345     entity.Property(e => e.Creationdate)
346         .HasColumnType("DATE")
347         .HasColumnName("CREATIONDATE");
348
349     entity.Property(e => e.MovieMovieId)
350         .HasColumnType("NUMBER")
351         .HasColumnName("MOVIE_MOVIE_ID");
352
353     entity.Property(e => e.Rating)
354         .HasPrecision(2)
355         .HasColumnName("RATING");
356
357     entity.Property(e => e.UserUserId)
358         .HasColumnType("NUMBER")
359         .HasColumnName("USER_USER_ID");
360
361     entity.HasOne(d => d.MovieMovie)
362         .WithMany(p => p.Reviews)
363         .HasForeignKey(d => d.MovieMovieId)
364         .OnDelete(DeleteBehavior.ClientSetNull)
365         .HasConstraintName("REVIEW_MOVIE_FK");

```

```

366         entity.HasOne(d => d.UserUser)
367             .WithMany(p => p.Reviews)
368             .HasForeignKey(d => d.UserUserId)
369             .OnDelete(DeleteBehavior.ClientSetNull)
370             .HasConstraintName("REVIEW_USER_FK");
371     });
372
373     modelBuilder.Entity<Room>(entity =>
374     {
375         entity.ToTable("ROOM");
376
377         entity.Property(e => e.RoomId)
378             .HasColumnType("NUMBER")
379             .HasColumnName("ROOM_ID");
380
381         entity.Property(e => e.Capacity)
382             .HasColumnType("NUMBER")
383             .HasColumnName("CAPACITY");
384
385         entity.Property(e => e.CinemaCinemaId)
386             .HasColumnType("NUMBER")
387             .HasColumnName("CINEMA_CINEMA_ID");
388
389         entity.Property(e => e.IsAvalible)
390             .IsRequired()
391             .HasMaxLength(1)
392             .IsUnicode(false)
393             .HasColumnName("IS_AVALIBLE")
394             .IsFixedLength(true);
395
396         entity.Property(e => e.Name)
397             .IsRequired()
398             .HasMaxLength(50)
399             .HasColumnName("NAME");
400
401         entity.HasOne(d => d.CinemaCinema)
402             .WithMany(p => p.Rooms)
403             .HasForeignKey(d => d.CinemaCinemaId)
404             .HasConstraintName("ROOM_CINEMA_FK");
405     });
406
407     modelBuilder.Entity<Screening>(entity =>
408     {
409         entity.ToTable("SCREENING");
410
411         entity.Property(e => e.ScreeningId)
412             .HasColumnType("NUMBER")
413             .HasColumnName("SCREENING_ID");
414
415         entity.Property(e => e.LanguageversionLvId)
416             .HasColumnType("NUMBER")
417             .HasColumnName("LANGUAGEVERSION_LV_ID");
418
419         entity.Property(e => e.MovieMovieId)
420             .HasColumnType("NUMBER")
421             .HasColumnName("MOVIE_MOVIE_ID");
422
423         entity.Property(e => e.Price)
424             .HasColumnType("FLOAT")
425             .HasColumnName("PRICE");
426
427         entity.Property(e => e.RoomRoomId)
428             .HasColumnType("NUMBER")
429             .HasColumnName("ROOM_ROOM_ID");
430
431         entity.HasOne(d => d.LanguageversionLv)
432             .WithMany(p => p.Screenings)
433             .HasForeignKey(d => d.LanguageversionLvId)
434             .OnDelete(DeleteBehavior.ClientSetNull)
435             .HasConstraintName("SCREENING_LANGUAGEVERSION_FK");
436
437         entity.HasOne(d => d.MovieMovie)

```

```

439         .WithMany(p => p.Screenings)
440         .HasForeignKey(d => d.MovieMovieId)
441         .OnDelete(DeleteBehavior.ClientSetNull)
442         .HasConstraintName("SCREENING_MOVIE_FK");
443
444     entity.HasOne(d => d.RoomRoom)
445         .WithMany(p => p.Screenings)
446         .HasForeignKey(d => d.RoomRoomId)
447         .OnDelete(DeleteBehavior.ClientSetNull)
448         .HasConstraintName("SCREENING_ROOM_FK");
449 });
450
451 modelBuilder.Entity<Search>(entity =>
452 {
453     entity.ToTable("SEARCH");
454
455     entity.Property(e => e.SearchId)
456         .HasColumnType("NUMBER")
457         .HasColumnName("SEARCH_ID");
458
459     entity.Property(e => e.Category)
460         .HasMaxLength(50)
461         .HasColumnName("CATEGORY");
462
463     entity.Property(e => e.Content)
464         .IsRequired()
465         .HasMaxLength(100)
466         .HasColumnName("CONTENT");
467
468     entity.Property(e => e.UserUserId)
469         .HasColumnType("NUMBER")
470         .HasColumnName("USER_USER_ID");
471
472     entity.HasOne(d => d.UserUser)
473         .WithMany(p => p.Searches)
474         .HasForeignKey(d => d.UserUserId)
475         .HasConstraintName("SEARCH_USER_FK");
476 });
477
478 modelBuilder.Entity<Ticket>(entity =>
479 {
480     entity.HasKey(e => new { e.UserUserId, e.ScreeningScreeningId })
481         .HasName("TICKET_PK");
482
483     entity.ToTable("TICKET");
484
485     entity.Property(e => e.UserUserId)
486         .HasColumnType("NUMBER")
487         .HasColumnName("USER_USER_ID");
488
489     entity.Property(e => e.ScreeningScreeningId)
490         .HasColumnType("NUMBER")
491         .HasColumnName("SCREENING_SCREENING_ID");
492
493     entity.Property(e => e.Date).HasColumnType("DATE");
494
495     entity.HasOne(d => d.ScreeningScreening)
496         .WithMany(p => p.Tickets)
497         .HasForeignKey(d => d.ScreeningScreeningId)
498         .OnDelete(DeleteBehavior.ClientSetNull)
499         .HasConstraintName("TICKET_SCREENING_FK");
500
501     entity.HasOne(d => d.UserUser)
502         .WithMany(p => p.Tickets)
503         .HasForeignKey(d => d.UserUserId)
504         .OnDelete(DeleteBehavior.ClientSetNull)
505         .HasConstraintName("TICKET_USER_FK");
506 });
507
508 modelBuilder.Entity<User>(entity =>
509 {
510     entity.ToTable("User");
511

```

```

512         entity.Property(e => e.UserId)
513             .HasColumnType("NUMBER")
514             .HasColumnName("USER_ID");
515
516         entity.Property(e => e.Email)
517             .IsRequired()
518             .HasMaxLength(80)
519             .HasColumnName("EMAIL");
520
521         entity.Property(e => e.Firstname)
522             .HasMaxLength(50)
523             .HasColumnName("FIRSTNAME");
524
525         entity.Property(e => e.Lastname)
526             .HasMaxLength(50)
527             .HasColumnName("LASTNAME");
528
529         entity.Property(e => e.Password)
530             .IsRequired()
531             .HasMaxLength(50)
532             .HasColumnName("PASSWORD");
533
534         entity.Property(e => e.Username)
535             .IsRequired()
536             .HasMaxLength(50)
537             .HasColumnName("USERNAME");
538     });
539
540     modelBuilder.Entity<Worker>(entity =>
541     {
542         entity.ToTable("WORKER");
543
544         entity.Property(e => e.WorkerId)
545             .HasColumnType("NUMBER")
546             .HasColumnName("WORKER_ID");
547
548         entity.Property(e => e.CinemaCinemaId)
549             .HasColumnType("NUMBER")
550             .HasColumnName("CINEMA_CINEMA_ID");
551
552         entity.Property(e => e.Name)
553             .IsRequired()
554             .HasMaxLength(50)
555             .HasColumnName("NAME");
556
557         entity.Property(e => e.Password)
558             .IsRequired()
559             .HasMaxLength(50)
560             .HasColumnName("PASSWORD");
561
562         entity.Property(e => e.Position)
563             .IsRequired()
564             .HasMaxLength(50)
565             .HasColumnName("POSITION");
566
567         entity.Property(e => e.Salary)
568             .HasPrecision(6)
569             .HasColumnName("SALARY");
570
571         entity.Property(e => e.Surname)
572             .IsRequired()
573             .HasMaxLength(50)
574             .HasColumnName("SURNAME");
575
576         entity.HasOne(d => d.CinemaCinema)
577             .WithMany(p => p.Workers)
578             .HasForeignKey(d => d.CinemaCinemaId)
579             .OnDelete(DeleteBehavior.ClientSetNull)
580             .HasConstraintName("WORKER_CINEMA_FK");
581     });

```