

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет по лабораторной работе № 3
Вариант 16 «Кинотеатр»

Выполнил:

ст. группы 950505
Киреев Ю.В.

Проверил:

Поденок Л. П.

Минск 2022

1. Задание

В лабораторной работе выполняется реализация схемы базы данных по ранее построенной реляционной схеме данных. Требуется сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД. Требуется заполнить таблицы данными с помощью оператора INSERT. Порядок выполнения лабораторной работы включает в себя следующие шаги:

Шаг 1. Создать в СУБД новую схему данных для хранения пользовательских объектов (см. часть 2).

Шаг 2. В этой новой схеме данных с помощью скрипта с запросами на языке DDL SQL реализовать таблицы, соответствующие реляционным отношениям схемы данных, полученной в лабораторной работе No2, с помощью одного (желательно) оператора CREATE TABLE для каждой таблицы в следующем порядке:

- реализовать простую структуру таблиц, включающую только набор столбцов с добавлением описаний первичного ключа;
- дополнить описание таблицы реализацией ограничений для описания внешних ключей; для внешних ключей установить свойства контроля целостности данных (каскадное удаление и обновление), если это возможно, в целевой СУБД;
- дополнить описание таблицы реализацией ограничений для описания бизнес-правил;
- дополнить описание таблицы реализацией комментариев для значимых элементов таблицы.

Шаг 3. Заполнить с помощью SQL-скрипта с использованием оператора INSERT таблицы строками данных для проверки правильного выбора первичных ключей и работоспособности ссылок между таблицами:

- строками данных сначала заполнять мастер-таблицы (или таблицы, которые НЕ ссылаются на другие таблицы);
- в каждую таблицу добавить 5 – 10 строк осмысленных данных;
- если не удастся добавить данные в таблицу по причине нарушения уникальности первичного ключа, то следует перепроверить описание этого первичного ключа и его смысл для реального мира;
- если не удастся добавить данные в таблицу по причине нарушения ссылочной целостности, то следует убедиться, что целевые данные существуют, иначе перепроверить описание внешнего ключа.

Шаг 4. Рассмотреть простые действия по изменению структуры таблицы (переименование столбца таблицы, добавление и удаление ограничений на столбец таблицы или всю таблицу) и реализовать их с помощью оператора ALTER TABLE.

Шаг 5. Создать временную таблицу с помощью оператора CREATE TABLE и удалить ее с помощью оператора DROP TABLE.

Шаг 6. Экспортировать результаты работы в SQL-скрипт, сравнить полученный скрипт со скриптами, созданными на этапах 2 и 3.

2. Выполнение работы

На основе ER-диаграмма, представленной на рисунке 2.1, в ходе выполнения лабораторной работы были сформулированы следующие SQL-запросы для создания таблиц базы данных:

```
CREATE TABLE IF NOT EXISTS public."EMPLOYEE"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 2 MINVALUE 2 MAXVALUE 2147483647 ),
    "Login" character varying(255) COLLATE pg_catalog."default" NOT
NULL,
    "Password" character varying(255) COLLATE pg_catalog."default" NOT
NULL,
    "Age" character varying(255) COLLATE pg_catalog."default",
    "FullName" character varying(255) COLLATE pg_catalog."default",
    "HALL_ID" integer,
    CONSTRAINT "EMPLOYEE_pkey" PRIMARY KEY ("ID"),
    CONSTRAINT "EMPLOYEE_HALL_ID_fkey" FOREIGN KEY ("HALL_ID")
        REFERENCES public."HALL" ("ID")
)

CREATE TABLE IF NOT EXISTS public."EMPLOYEE-POSITION"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "EMPLOYEE_ID" integer,
    "POSITION_ID" integer,
    CONSTRAINT "EMPLOYEE-POSITION_pkey" PRIMARY KEY ("ID"),
    CONSTRAINT "EMPLOYEE-POSITION_EMPLOYEE_ID_fkey" FOREIGN KEY
("EMPLOYEE_ID")
        REFERENCES public."EMPLOYEE" ("ID")
    CONSTRAINT "EMPLOYEE-POSITION_POSITION_ID_fkey" FOREIGN KEY
("POSITION_ID")
        REFERENCES public."POSITION" ("ID")
)

CREATE TABLE IF NOT EXISTS public."FILM"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "Name" character varying(255) COLLATE pg_catalog."default",
    "Description" text COLLATE pg_catalog."default",
    "SESSION_ID" integer,
    CONSTRAINT "FILM_pkey" PRIMARY KEY ("ID"),
    CONSTRAINT "FILM_SESSION_ID_fkey" FOREIGN KEY ("SESSION_ID")
        REFERENCES public."SESSION" ("ID")
)

CREATE TABLE IF NOT EXISTS public."HALL"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "Name" character varying(255) COLLATE pg_catalog."default",
    "SESSION_ID" integer,
```

```

        CONSTRAINT "HALL_pkey" PRIMARY KEY ("ID"),
        CONSTRAINT "HALL_SESSION_ID_fkey" FOREIGN KEY ("SESSION_ID")
            REFERENCES public."SESSION" ("ID")
    )

CREATE TABLE IF NOT EXISTS public."HALLROW"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "Number" character varying(255) COLLATE pg_catalog."default",
    "Count" integer,
    "HALL_ID" integer,
    "OCCUPANCY_ID" integer,
    CONSTRAINT "HALLROW_pkey" PRIMARY KEY ("ID"),
    CONSTRAINT "HALLROW_HALL_ID_fkey" FOREIGN KEY ("HALL_ID")
        REFERENCES public."HALL" ("ID") ,
    CONSTRAINT "HALLROW_OCCUPANCY_ID_fkey" FOREIGN KEY
("OCCUPANCY_ID")
        REFERENCES public."OCCUPANCY" ("ID")
)

CREATE TABLE IF NOT EXISTS public."OCCUPANCY"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "Place" character varying(255) COLLATE pg_catalog."default",
    "Row" character varying(255) COLLATE pg_catalog."default",
    CONSTRAINT "OCCUPANCY_pkey" PRIMARY KEY ("ID"),
    CONSTRAINT "OCCUPANCY_SESSION_ID_fkey" FOREIGN KEY ("SESSION_ID")
        REFERENCES public."SESSION" ("ID")
)

CREATE TABLE IF NOT EXISTS public."POSITION"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "Name" character varying(255) COLLATE pg_catalog."default",
    "Salary" bigint,
    CONSTRAINT "POSITION_pkey" PRIMARY KEY ("ID")
)

CREATE TABLE IF NOT EXISTS public."SESSION"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 ),
    "Time" time with time zone,
    "AmountPlaces" integer,
    "OCCUPANCY_ID" integer,
    CONSTRAINT "SESSION_pkey" PRIMARY KEY ("ID")
)

CREATE TABLE IF NOT EXISTS public."TICKET"
(
    "ID" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1
START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    "Place" character varying(255) COLLATE pg_catalog."default",

```

```
"Row" character varying(255) COLLATE pg_catalog."default",
"Price" bigint,
"EMPLOYEE_ID" integer,
"SESSION_ID" integer,
"HALLROW_ID" integer,
CONSTRAINT "TICKET_pkey" PRIMARY KEY ("ID"),
CONSTRAINT "TICKET_EMPLOYEE_ID_fkey" FOREIGN KEY ("EMPLOYEE_ID")
    REFERENCES public."EMPLOYEE" ("ID"),
CONSTRAINT "TICKET_HALLROW_ID_fkey" FOREIGN KEY ("HALLROW_ID")
    REFERENCES public."HALLROW" ("ID"),
CONSTRAINT "TICKET_SESSION_ID_fkey" FOREIGN KEY ("SESSION_ID")
    REFERENCES public."SESSION" ("ID")
)
```

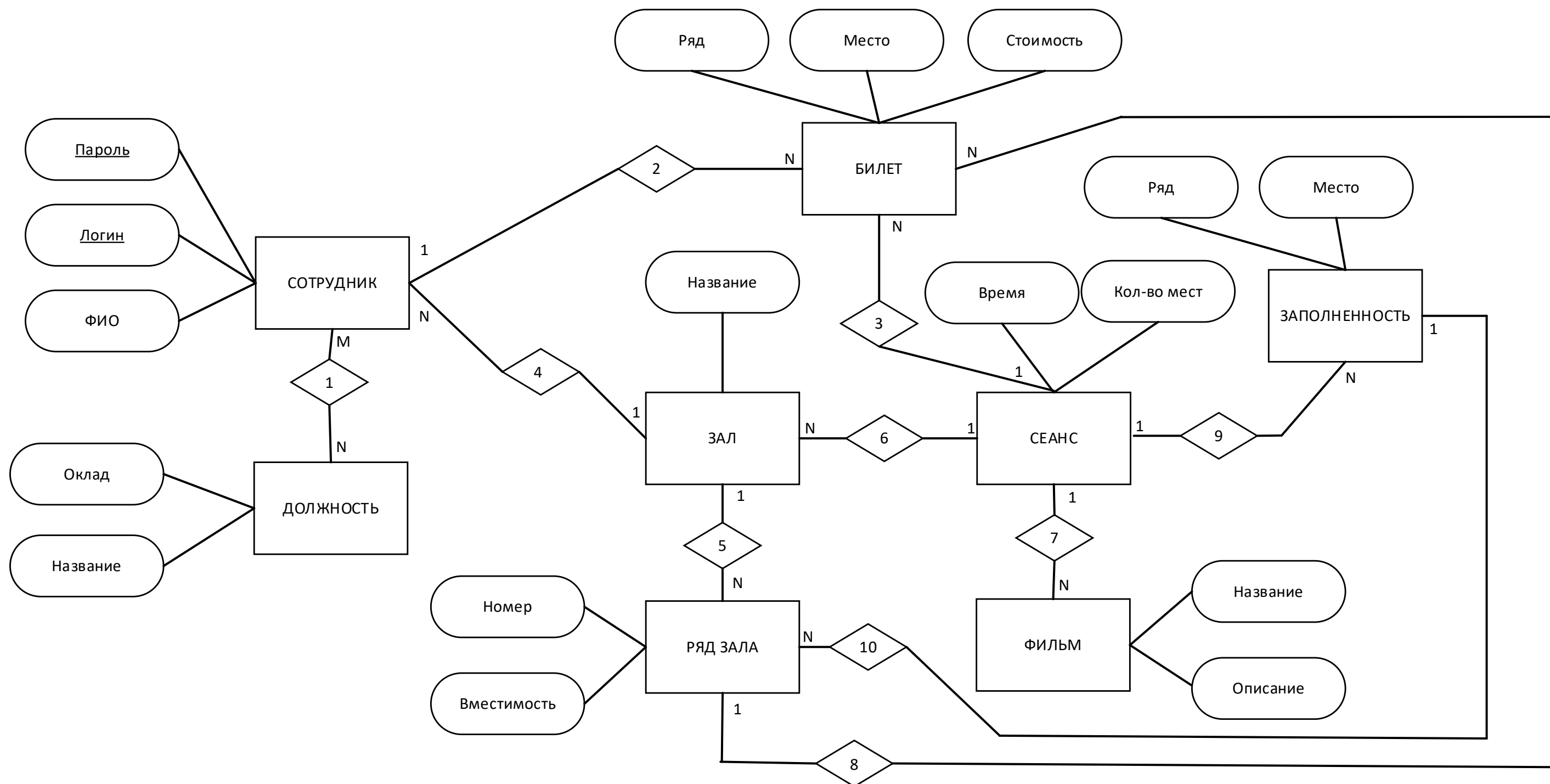


Рисунок 2.1 – ER-диаграмма “Кинотеатр”

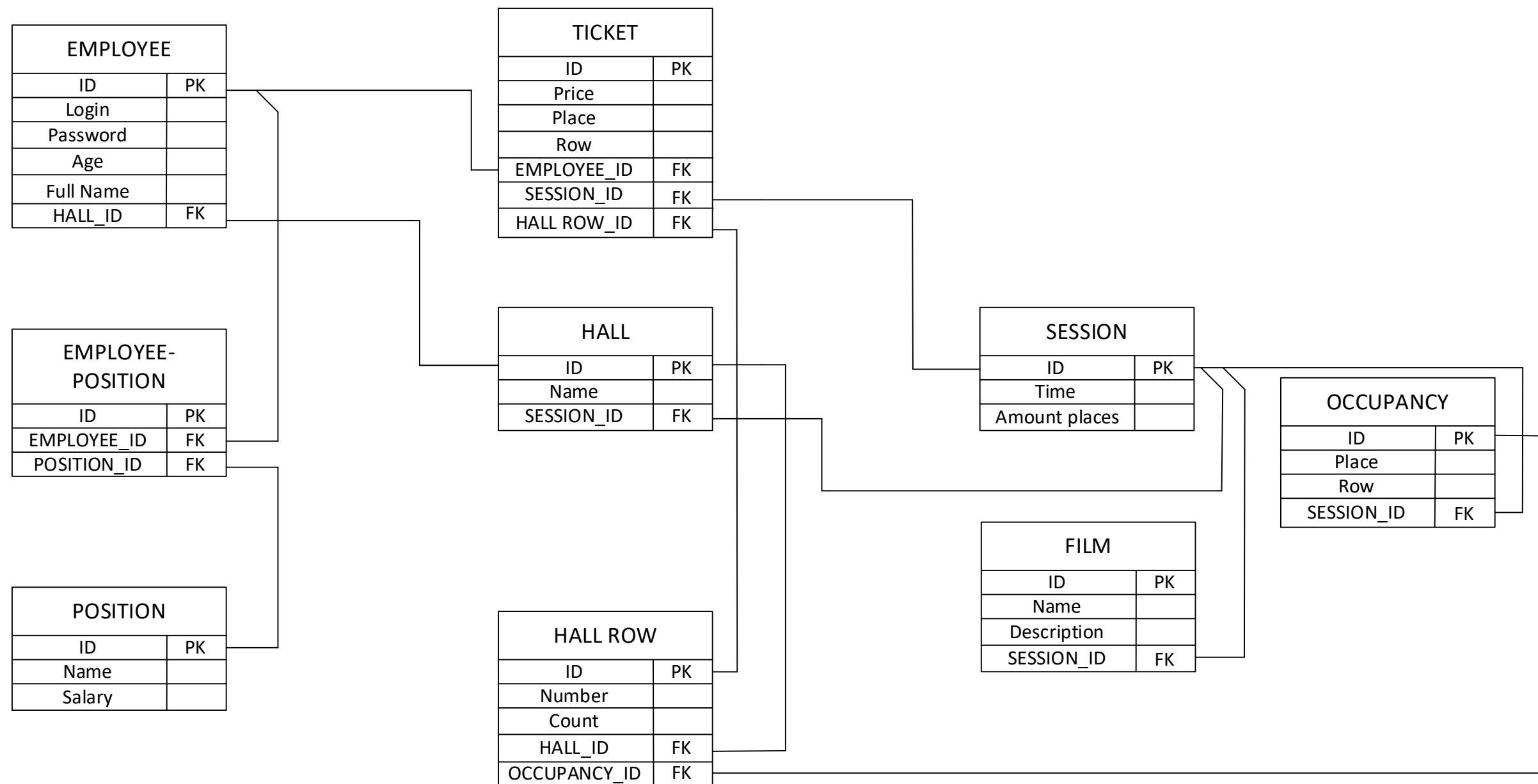


Рисунок 2.2 – Реляционная диаграмма “Кинотеатр”