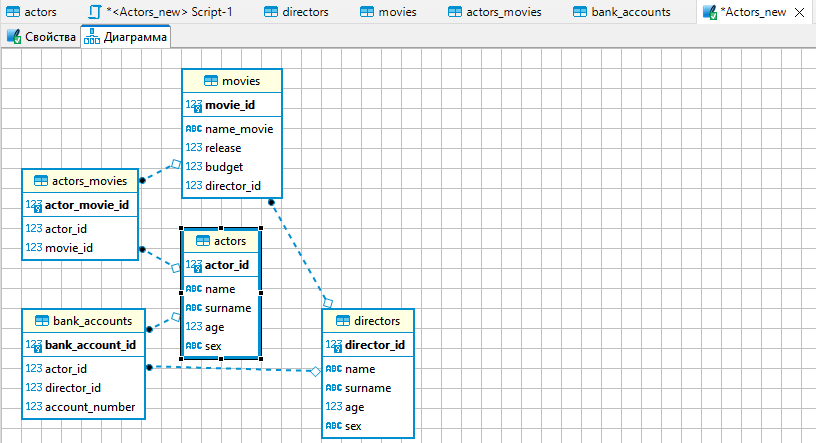
**Задачи на выборку из нескольких таблиц и не только**

1. Создать таблицы с данными согласно спроектированной схеме. Заполнить предоставленными данными. 
2. Добавить новое поле finance INTEGER в таблицу bank\_accounts значение задать NULL.   
     
   **ALTER** **table** bank\_accounts **ADD** **COLUMN** finance **INTEGER** **IS** **NULL**
3. Вывести первых 10 режиссёров, которые сняли высокобюджетные фильмы. Режиссёры не должны повторяться.  
     
   **SELECT** **DISTINCT** d.name, d.surname **FROM** directors d **JOIN** movies m **USING** (director\_id) **ORDER** **BY** m.budget **DESC** **LIMIT** 10
4. Вывести актёров и режиссёров, которые не участвовали не в одном из фильмов.  
     
   **SELECT** \* **FROM** actors **WHERE** actor\_id **NOT** **IN** (**SELECT** **DISTINCT** actor\_id **FROM** actors\_movies am) **UNION** **SELECT** \* **FROM** directors d **WHERE** director\_id **NOT** **IN** (**SELECT** **DISTINCT** director\_id **FROM** movies m)
5. Вывести все фильмы, а так же всех актёров кассовые сборы которых превысили 150000000.  
     
   **SELECT** name\_movie, GROUP\_CONCAT(a.name || ' ' || a.surname) **FROM** movies m

**LEFT** **JOIN** actors\_movies am **USING** (movie\_id)

**LEFT** **JOIN** actors a **USING** (actor\_id)

**WHERE** m.budget>150000000 **AND** actor\_id **NOT** **NULL** **GROUP** **BY** movie\_id

1. Вывести всех режиссёров которые снимали фильмы до 2000 года. Режиссёры не должны повторятся  
     
   **SELECT** **DISTINCT** d.name, d.surname **FROM** directors d **JOIN** movies m **USING** (director\_id) **WHERE** m."release" <2000
2. Добавить фильмы для актёров и режиссёров, у которых нет зависимости в таблице movies. Не забыть сделать изменения в таблице actors\_movies.  
   Добавлены
3. Добавить колонку rating к фильмам. Задать значения рейтинга фильмов.   
    **ALTER** **table** movies **ADD** **COLUMN** rating **REAL NOT NULL**
4. Вывести режиссёров и фильмы с рейтингом ниже среднего до 2000 года  
    **SELECT** name\_movie, GROUP\_CONCAT(d.name || ' ' || d.surname) **as** director

**FROM** movies m **LEFT** **JOIN** directors d **USING** (director\_id)

**WHERE** rating<(**SELECT** **AVG**(rating) **FROM** movies m)

**AND** "release" <2000 **GROUP** **BY** movie\_id

1. Вывести всех актёров, которые знакомы с 2-мя и более другими актёрами.  
   **SELECT** a.name, a.surname

**FROM** actors a

**WHERE** a.actor\_id **IN** (

**SELECT** am1.actor\_id

**FROM** actors\_movies am1

**JOIN** actors\_movies am2 **ON** am1.movie\_id = am2.movie\_id **AND** am1.actor\_id <> am2.actor\_id

**GROUP** **BY** am1.actor\_id

**HAVING** **COUNT**(**DISTINCT** am2.actor\_id) >= 2

)

**Дополнительные задачи**

1. Вывести первых 5 самых высокооплачиваемых актёров.   
     
   **SELECT** name, surname, finance **FROM** bank\_accounts **JOIN** actors a **USING** (actor\_id) **ORDER** **BY** finance **DESC** **LIMIT** 5
2. Вывести всех режиссёров и актёров которые были задействованы только в 1 фильме (если просто по 1 фильму, а не совместно только в одном)  
     
   **SELECT** name\_movie, a.name, a.surname, 'actor' **as** profession

**FROM** movies m

**JOIN** actors\_movies am **USING** (movie\_id)

**JOIN** actors a **USING** (actor\_id)

**GROUP** **BY** a.actor\_id **HAVING** **COUNT**(a.actor\_id)=1

**UNION**

**SELECT** name\_movie, d.name, d.surname, 'director' **as** proffession

**FROM** movies m

**JOIN** directors d **USING** (director\_id)

**GROUP** **BY** d.director\_id **HAVING** **COUNT**(d.director\_id)=1

1. В поле finance таблицы bank\_accounts внести информацию о доходах актёров. Доход считается с кассового сбора фильма. Если доход с фильма составил более 150000000 это 20%, от 150000000 до 100000000 – 15 %, если менее 100 миллионов 10%. Запрос на изменение таблицы bank\_acounts должен быть выполнен с помощью подзапроса, который производит подсчёт суммы всех гонораров за фильмы.

**UPDATE** bank\_accounts **AS** ba

**SET** finance = (

**SELECT** **SUM**(**CASE**

**WHEN** budget > 150000000 **THEN** budget \* 0.20

**WHEN** budget >= 100000000 **AND** budget <= 150000000 **THEN** budget \* 0.15

**ELSE** budget \* 0.10

**END**) **AS** fee

**FROM** (

**SELECT** m.movie\_id, m.budget, am.actor\_id

**FROM** movies m

**JOIN** actors\_movies am **USING** (movie\_id)

**WHERE** am.actor\_id **IS** **NOT** **NULL**

) **AS** fin\_sum

**WHERE** ba.actor\_id = fin\_sum.actor\_id

);