**Нуритдинходжаева Алина С20-712**

**Лабораторная работа №3**

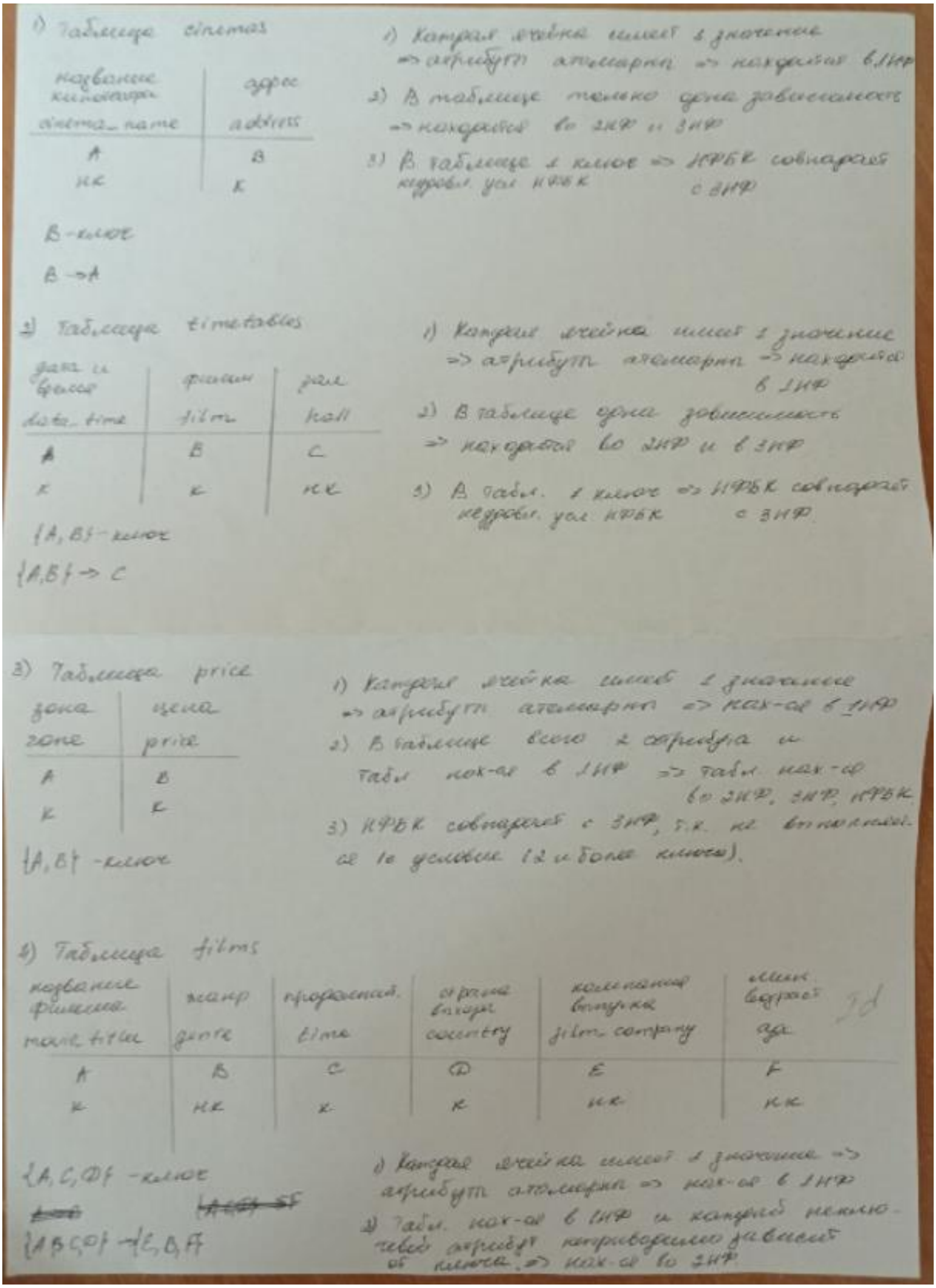
**Вариант №17**

**Общая часть заданий**

**--Задание 1**

```
CREATE TABLE actions (
   position VARCHAR2(64 CHAR) NOT NULL,
   task VARCHAR2(128 CHAR),
   films_id INTEGER NOT NULL CHECK (films_id >1),
   people_id INTEGER NOT NULL CHECK (people_id >1));

ALTER TABLE actions ADD CONSTRAINT actions_pk PRIMARY KEY (films_id, people_id);

ALTER TABLE actions ADD CONSTRAINT actions_position_un UNIQUE (position);

CREATE TABLE cinemas (
   cinema_name VARCHAR2(64 CHAR),
   address     VARCHAR2(128 CHAR) NOT NULL);

ALTER TABLE cinemas ADD CONSTRAINT cinemas_pk PRIMARY KEY (address);

CREATE TABLE films (
   movie_title VARCHAR2(64 CHAR) NOT NULL,
   genre VARCHAR2(32 CHAR),
   time INTEGER NOT NULL,
   country VARCHAR2(32 CHAR) NOT NULL,
   film_company VARCHAR2(128 CHAR),
   age INTEGER,
   id INTEGER NOT NULL CHECK (id > 0));

ALTER TABLE films ADD CONSTRAINT films_pk PRIMARY KEY (id);

ALTER TABLE films
   ADD CONSTRAINT fmttc UNIQUE (movie_title, time, country);

CREATE TABLE people (
   last_name VARCHAR2(64 CHAR) NOT NULL,
   first_name VARCHAR2(64 CHAR) NOT NULL,
   birthday DATE NOT NULL,
   id INTEGER NOT NULL);

ALTER TABLE people ADD CONSTRAINT people_pk PRIMARY KEY (id);

ALTER TABLE people
   ADD CONSTRAINT pln UNIQUE (last_name, birthday, first_name);

CREATE TABLE price (
   zone VARCHAR2(32 CHAR) NOT NULL,
   timetables_date_time DATE NOT NULL,
   timetables_film VARCHAR2(64 CHAR) NOT NULL,
   price INTEGER NOT NULL);

ALTER TABLE price ADD CONSTRAINT price_pk PRIMARY KEY (zone,
price);

CREATE TABLE timetables (
   date_time DATE NOT NULL,
   film VARCHAR2(64 CHAR) NOT NULL,
   hall INTEGER,
   cinemas_address VARCHAR2(128 CHAR) NOT NULL,
   films_id INTEGER NOT NULL);

ALTER TABLE timetables ADD CONSTRAINT timetables_pk PRIMARY KEY (date_time, film);

ALTER TABLE actions
   ADD CONSTRAINT actions_films_fk FOREIGN KEY ( films_id )
     REFERENCES films (id);

ALTER TABLE actions
   ADD CONSTRAINT actions_people_fk FOREIGN KEY ( people_id )
     REFERENCES people (id);

ALTER TABLE price
   ADD CONSTRAINT price_timetables_fk FOREIGN KEY (timetables_date_time, timetables_film)
     REFERENCES timetables (date_time, film);

ALTER TABLE timetables
   ADD CONSTRAINT timetables_cinemas_fk FOREIGN KEY (cinemas_address)
     REFERENCES cinemas (address);

ALTER TABLE timetables
   ADD CONSTRAINT timetables_films_fk FOREIGN KEY (films_id)
     REFERENCES films (id);
```

## --Задание 2



1) Таблица cinemas

| название кинотеатра<br>cinema_name | адрес<br>address |
|---|---|
| A | B |
| нк | к |

В-ключ

B → A

1) Каждая ячейка имеет 1 значение => атрибуты атомарны => находится в 1НФ

2) В таблице только одна зависимость => находится во 2НФ и 3НФ

3) В таблице 1 ключ => НФБК совпадает с 3НФ
недовып. усл НФБК

2) Таблица timetables

| дата и время<br>date_time | фильм<br>film | зал<br>hall |
|---|---|---|
| A | B | C |
| к | к | нк |

{A, B}-ключ

{A,B} → C

1) Каждая ячейка имеет 1 значение => атрибуты атомарны => находятся в 1НФ

2) В таблице одна зависимость => находятся во 2НФ и в 3НФ

3) В табл. 1 ключ => НФБК совпадает с 3НФ.
недовып. усл НФБК

3) Таблица price

| зона<br>zone | цена<br>price |
|---|---|
| A | B |
| к | к |

{A, B} -ключ

1) Каждая ячейка имеет 1 значение => атрибуты атомарны => нах-ся в 1НФ

2) В таблице всего 2 атрибута и табл нах-ся в 1НФ => табл нах-ся во 2НФ, 3НФ, НФБК

3) НФБК совпадает с 3НФ, т.к. не выполняет-ся 1е условие (2 и более ключа).

4) Таблица films

| название фильма<br>movie_title | жанр<br>genre | продолжит.<br>time | страна выпуска<br>country | компания выпуска<br>film_company | мин возраст<br>age |
|---|---|---|---|---|---|
| A | B | C | D | E | F |
| к | нк | к | к | нк | нк |

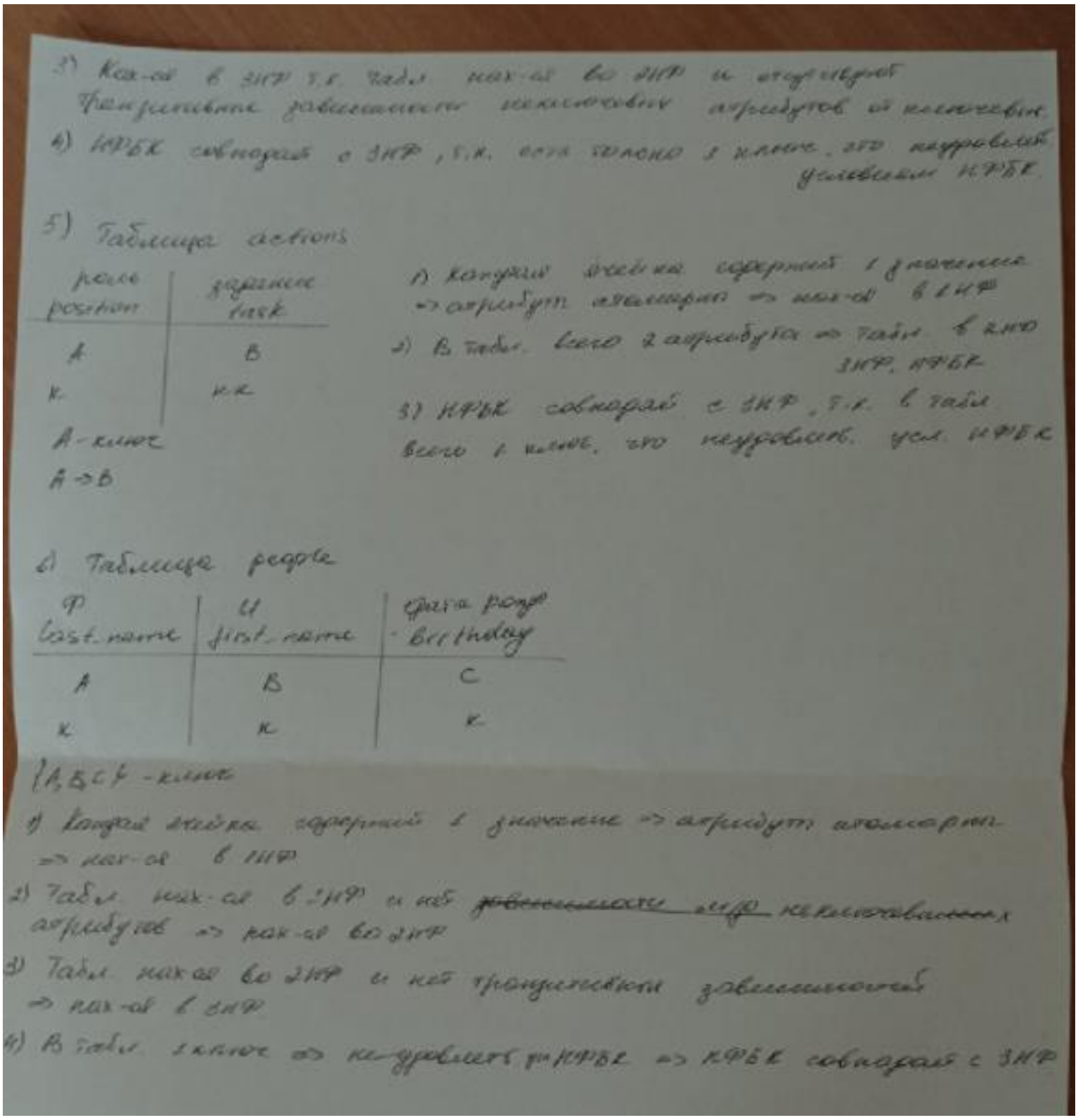
{A, C, D} -ключ

~~[illegible]~~ ~~{A,C,D} → F~~

{A,B,C,D} → {E,B,F}

1) Каждая ячейка имеет 1 значение => атрибуты атомарны => нах-ся в 1НФ

2) Табл. нах-ся в 1НФ и каждый неключевой атрибут неприводимо зависит от ключа => нах-ся во 2НФ



3) Нах-ся в 3НФ т.к. табл. нах-ся во 2НФ и отсутствуют транзитивные зависимости неключевых атрибутов от ключевых.

4) НФБК совпадает с 3НФ, т.к. есть только 1 ключ, что неудовлетв. условиям НФБК.

5) Таблица actions

| роль<br>position | задание<br>task |
|---|---|
| A | B |
| к | н.к |

A-ключ

A->B

1) Каждая ячейка содержит 1 значение => атрибут атомарен => нах-ся в 1НФ

2) В табл. всего 2 атрибута => табл. в 2НФ 3НФ, НФБК

3) НФБК совпадает с 3НФ, т.к. в табл. всего 1 ключ, что неудовлетв. усл. НФБК

6) Таблица people

| Ф<br>last_name | И<br>first_name | дата рожд<br>- birthday |
|---|---|---|
| A | B | C |
| к | к | к |

{A,B,C} - ключ

1) Каждая ячейка содержит 1 значение => атрибут атомарен => нах-ся в 1НФ

2) Табл. нах-ся в 1НФ и нет ~~зависимостей~~ ~~мф~~ неключевых атрибутов => нах-ся во 2НФ

3) Табл. нах-ся во 2НФ и нет транзитивных зависимостей => нах-ся в 3НФ

4) В табл. 1 ключ => не удовлет. усл НФБК => НФБК совпадает с 3НФ

**--Задание 3**

--заполнение таблицы cinemas

INSERT INTO cinemas

VALUES ('Формула Кино Чертаново', 'м.Чертаново, ТЦ "Авентура"');

INSERT INTO cinemas

VALUES ('Кронверк Синема Облака', 'м.Домодедово, ТРК "Облака"');

INSERT INTO cinemas

VALUES ('Синема Парк Ривьера', 'м.Автозаводская, ТРЦ "Ревьера"');

INSERT INTO cinemas

```
VALUES ('Синема Парк Европейский', 'м.Киевская, "Европейский"');
INSERT INTO cinemas
VALUES ('КИНО ОККО', 'м.Выставочная, ТЦ "Афимолл-сити"');

--заполнение таблицы films
INSERT INTO films
VALUES ('Первый оскар', 'драма', 114, 'Россия', 'СТВ', 12, 1);
INSERT INTO films
VALUES ('Казнь', 'триллер', 130, 'Канада', 'A24', 18, 2);
INSERT INTO films
VALUES ('Стикер', 'детектив', 117, 'Россия', 'ТаББак', 16, 3);
INSERT INTO films
VALUES ('Белый снег', 'драма', 133, 'Россия', 'СТВ', 6, 4);
INSERT INTO films
VALUES ('Брат', 'боевик', 100, 'Россия', 'ТаББак', 12, 5);
INSERT INTO films
VALUES ('Терминатор', 'фантастика', 107, 'США', 'Cinema 84', 16, 6);

--заполнение таблицы actions
INSERT INTO actions (position, films_id, people_id)
VALUES ('режисер', 5, 10);
INSERT INTO actions (position, films_id, people_id)
VALUES ('суфлер', 4, 11);
INSERT INTO actions (position, films_id, people_id)
VALUES ('гример', 2, 9);
INSERT INTO actions (position, films_id, people_id)
VALUES ('актер', 2, 13);
INSERT INTO actions (position, films_id, people_id)
VALUES ('сценарист', 5, 12);
```

```
INSERT INTO actions (position, films_id, people_id)
VALUES ('дублер', 4, 10);

--заполнение таблицы people
INSERT INTO people
VALUES ('Балабанов', 'Алексей', '12.12.1985', 12);
INSERT INTO people
VALUES ('Хомерики', 'Николай', '25.02.1976', 11);
INSERT INTO people
VALUES ('Баран', 'Дмитрий', '06.05.1990', 13);
INSERT INTO people
VALUES ('Кваталия', 'Ладо', '21.08.1969', 10);
INSERT INTO people
VALUES ('Лернер', 'Майкл', '30.06.1981', 9);
INSERT INTO people
VALUES ('Кошмал', 'Анна', '26.09.2000', 14);

--заполнение таблицы timetables
INSERT INTO timetables
VALUES ('01.05.2022', 'Первый оскар', 1, 'м.Выставочная, ТЦ "Афимолл-сити"', 1 );
INSERT INTO timetables
VALUES ('01.05.2022', 'Казнь', 2, 'м.Выставочная, ТЦ "Афимолл-сити"', 2);
INSERT INTO timetables
VALUES ('01.05.2022', 'Брат', 1, 'м.Автозаводская, ТРЦ "Ревьера"', 5);
INSERT INTO timetables
VALUES ('01.05.2022', 'Стикер', 2, 'м.Киевская, "Европейский"', 3);
INSERT INTO timetables
VALUES ('30.04.2022', 'Белый снег', 3, 'м.Киевская, "Европейский"', 4 );
```

```
INSERT INTO timetables
VALUES ('29.04.2022', 'Терминатор', 1, 'м.Киевская, "Европейский"', 6);

--заполнение таблицы price
INSERT INTO price
VALUES ('красная', '30.04.2022', 'Белый снег', 400);
INSERT INTO price
VALUES ('желтая', '30.04.2022', 'Белый снег', 350);
INSERT INTO price
VALUES ('синяя', '30.04.2022', 'Белый снег', 300);
INSERT INTO price
VALUES ('красная', '01.05.2022', 'Брат', 500);
INSERT INTO price
VALUES ('желтая', '01.05.2022', 'Брат', 450);
INSERT INTO price
VALUES ('синяя', '01.05.2022', 'Брат', 400);
```

**Задания по вариантам**

**--Задание 0**

```
SELECT last_name FROM (
SELECT last_name, COUNT(people_id) AS que
FROM actions INNER JOIN people ON people_id=id
GROUP BY last_name)
WHERE que = (SELECT MAX(que) FROM (SELECT last_name,
COUNT(people_id) AS que FROM actions INNER JOIN people ON people_id=id
GROUP BY last_name));
```

**--Задание 1**

```
SELECT film_company, COUNT(movie_title) FROM films

GROUP BY film_company;
```

**--Задание 2**

```
SELECT cinema_name

FROM cinemas INNER JOIN timetables ON address=cinemas_address

WHERE REGEXP_LIKE(date_time, '\d\d\.04\.22') AND REGEXP_LIKE(film, 'Терминатор');
```

**--Задание 3**

```
DELETE FROM people

WHERE last_name = (SELECT last_name

FROM people LEFT JOIN actions ON people_id=id WHERE position IS NULL);
```

**--Задание 4**

```
UPDATE price SET price = 2 * price

WHERE ((SELECT date_time

FROM (cinemas INNER JOIN timetables ON address = cinemas_address)
WHERE date_time >= sysdate
AND cinema_name = 'Синема Парк Европейский') = timetables_date_time);


SELECT * FROM price;
```

**--Задание 5**

```
ALTER TABLE cinemas ADD net VARCHAR2 (64 CHAR);
```

**--Задание 6**

```
ALTER TABLE price ADD CHECK (price > 50);
```