

1. В таблице **products** хранятся товары с текущими ценами, а в **old_prices** — старые цены на эти же товары. Если товар добавили недавно, то в **old_prices** данных о нем нет.

Получите из таблицы **products** id, название и текущую стоимость товаров, цены которых были изменены. Данные отсортируйте по id.

```
SELECT product_id, products.price
FROM products JOIN old_prices ON products.id = old_prices.product_id
WHERE products.price != old_prices.price
ORDER BY product_id;
```

	product_id	price
1	5	36
2	6	46
3	7	45
4	10	56
5	16	38
6	21	38

2. В таблице **people** содержится список всех людей, а в **suspects** — подозреваемых. Получите из **people** всех подозреваемых основываясь на данных из таблицы **suspects**.

Данные должны быть отсортированы по id.

```
SELECT people.* FROM people, suspects
WHERE people.last_name = suspects.ln AND people.first_name = suspects.fn
AND people.age = suspects.age
ORDER BY id;
```

	id	first_name	last_name	age	sex
1	4	Денис	Кац	22	m
2	9	Анатолий	Дейчман	21	m
3	13	Анастасия	Татаренкова	21	w
4	15	Денис	Кац	22	m

3. В таблицах **first_names** и **last_names** хранятся все разрешенные в системе имена и фамилии, а в таблице **people** — все зарегистрированные пользователи.

Получите из **people** всех людей, у которых неверные имена или фамилии.
Выведите id, имя и фамилию. Данные отсортируйте по фамилии.

```
SELECT id, first_name, last_name FROM people
WHERE CONCAT(people.first_name, people.last_name) NOT IN (
    SELECT CONCAT(first_names.first_name, last_names.last_name)
    FROM first_names, last_names
)
ORDER BY last_name;
```

	id	first_name	last_name
1	5	Сорокина	Алена
2	14	Валерий	Ковальчук
3	7	Ярослава	Сорокина
4	10	Юлиана	Фёдорова

4. На главной странице сайта выводится ТОП5 компьютерных игр. Чтобы вывод был разнообразным:
 1. Берут по две самых популярных игры из 5 категорий.
 2. Из полученной десятки оставляют пять лучших игр.
 3. Итоговую пятерку сортируют по рейтингу в **прямом** порядке.

Строки с одинаковым рейтингом отсортируйте по id.

Получите из таблицы **games** данные для вывода на главную страницу сайта для категорий: 1 - Action, 2 - RPG, 3 - Adventure, 4 - Strategy и 5 - Shooter. Выведите поля **id**, **name**, **rating** и **genre**, где **genre** — название категории.

```
(
  (
    SELECT id, NAME, rating, 'Action' AS genre FROM games
    WHERE category_id = 1
    ORDER BY rating DESC
    LIMIT 2
  )
  UNION
  (
    SELECT id, NAME, rating, 'RPG' AS genre FROM games
    WHERE category_id = 2
    ORDER BY rating DESC
    LIMIT 2
  )
  UNION
  (
    SELECT id, NAME, rating, 'Adventure' AS genre FROM games
    WHERE category_id = 3
    ORDER BY rating DESC
    LIMIT 2
  )
  UNION
  (
    SELECT id, NAME, rating, 'Strategy' AS genre FROM games
    WHERE category_id = 4
    ORDER BY rating DESC
    LIMIT 2
  )
  UNION
  (
    SELECT id, NAME, rating, 'Shooter' AS genre FROM games
    WHERE category_id = 5
    ORDER BY rating DESC
    LIMIT 2
  )
  ORDER BY rating DESC LIMIT 5
)
```

```
ORDER BY rating;
```

	id	NAME	rating	genre
1	16	Metro: Last Light	9.25	Shooter
2	19	Warcraft III: Reign of Chaos	9.29	Strategy
3	14	Uncharted 4: A Thief's End	9.33	Adventure
4	4	The Last of Us	9.45	Adventure
5	1	The Witcher 3: Wild Hunt	9.61	RPG

5. Для хранения транзакций в базе данных используется 3 таблицы: **bank_transactions**, **cashbox_transactions** и **paypal_transactions**.

Выведите дату (колонка date), сумму и тип последних трех транзакций. Итоговые данные отсортируйте по дате в хронологическом порядке. В качестве типа используйте bank, cash и paypal в зависимости от таблицы. Тип выведите в поле payment_type.

```
SELECT * FROM(
    SELECT * FROM(
        SELECT *, 'bank' AS payment_type FROM bank_transactions
        UNION
        SELECT *, 'cash' AS payment_type FROM cashbox_transactions
        UNION
        SELECT *, 'paypal' AS payment_type FROM paypal_transactions
    ) AS sum
    ORDER BY date DESC
    LIMIT 3
) AS inverse
ORDER BY date;
```

	id	client_id	date	amount	payment_type
1	3	98	2017-02-16 09:00:30	1000.00	cash
2	4	17	2017-02-17 08:45:29	5400.50	paypal
3	3	56	2017-02-17 08:45:30	4050.24	paypal

6. В таблицах **first_names** и **last_names** хранятся все доступные в системе имена и фамилии. Получите все возможные комбинации фамилий и имен. Фамилии мужского пола могут сочетаться только с мужскими именами, тоже касается и женских фамилий и имен.

В итоговой таблице выведите фамилии, имена и пол. Данные отсортируйте сперва по полу, а затем по фамилии и имени.

```
SELECT
    f.id AS first_name_id,
    f.first_name,
    f.sex AS first_name_sex,
    l.id AS last_name_id,
    l.last_name,
    l.sex AS last_name_sex
FROM
    first_names f
JOIN
    last_names l
ON
    f.sex = l.sex
ORDER BY
    f.sex,
    l.last_name,
    f.first_name;
```

	first_name_id	first_name	first_name_sex	last_name_id	last_name	last_name_sex
1	8	Анатолий	м	7	Белый	м
2	10	Андрей	м	7	Белый	м
3	13	Валерий	м	7	Белый	м
4	6	Василий	м	7	Белый	м
5	4	Денис	м	7	Белый	м
6	1	Дмитрий	м	7	Белый	м
7	3	Никита	м	7	Белый	м
8	7	Ярослав	м	7	Белый	м
9	8	Анатолий	м	8	Дейчман	м
10	10	Андрей	м	8	Дейчман	м
11	13	Валерий	м	8	Дейчман	м
12	6	Василий	м	8	Дейчман	м
13	4	Денис	м	8	Дейчман	м
14	1	Дмитрий	м	8	Дейчман	м
15	3	Никита	м	8	Дейчман	м
16	7	Ярослав	м	8	Дейчман	м
17	8	Анатолий	м	4	Кац	м
18	10	Андрей	м	4	Кац	м
19	13	Валерий	м	4	Кац	м
20	6	Василий	м	4	Кац	м
21	4	Денис	м	4	Кац	м
22	1	Дмитрий	м	4	Кац	м
23	3	Никита	м	4	Кац	м
24	7	Ярослав	м	4	Кац	м
25	8	Анатолий	м	13	Редько	м
26	10	Андрей	м	13	Редько	м
27	13	Валерий	м	13	Редько	м
28	6	Василий	м	13	Редько	м
29	4	Денис	м	13	Редько	м
30	1	Дмитрий	м	13	Редько	м
31	3	Никита	м	13	Редько	м
32	7	Ярослав	м	13	Редько	м
33	8	Анатолий	м	10	Ройзман	м
34	10	Андрей	м	10	Ройзман	м
35	13	Валерий	м	10	Ройзман	м

36	6 Василий	m	10 Ройзман	m
37	4 Денис	m	10 Ройзман	m
38	1 Дмитрий	m	10 Ройзман	m
39	3 Никита	m	10 Ройзман	m
40	7 Ярослав	m	10 Ройзман	m
41	8 Анатолий	m	6 Сорокин	m
42	10 Андрей	m	6 Сорокин	m
43	13 Валерий	m	6 Сорокин	m
44	6 Василий	m	6 Сорокин	m
45	4 Денис	m	6 Сорокин	m
46	1 Дмитрий	m	6 Сорокин	m
47	3 Никита	m	6 Сорокин	m
48	7 Ярослав	m	6 Сорокин	m
49	8 Анатолий	m	3 Трутнев	m
50	10 Андрей	m	3 Трутнев	m
51	13 Валерий	m	3 Трутнев	m
52	6 Василий	m	3 Трутнев	m
53	4 Денис	m	3 Трутнев	m
54	1 Дмитрий	m	3 Трутнев	m
55	3 Никита	m	3 Трутнев	m
56	7 Ярослав	m	3 Трутнев	m
57	8 Анатолий	m	1 Федьков	m
58	10 Андрей	m	1 Федьков	m
59	13 Валерий	m	1 Федьков	m
60	6 Василий	m	1 Федьков	m
61	4 Денис	m	1 Федьков	m
62	1 Дмитрий	m	1 Федьков	m
63	3 Никита	m	1 Федьков	m
64	7 Ярослав	m	1 Федьков	m
65	5 Алена	w	2 Иванова	w
66	12 Анастасия	w	2 Иванова	w
67	2 Светлана	w	2 Иванова	w
68	11 София	w	2 Иванова	w
69	9 Юлия	w	2 Иванова	w

70	5 Алена	w	5 Сорокина	w
71	12 Анастасия	w	5 Сорокина	w
72	2 Светлана	w	5 Сорокина	w
73	11 София	w	5 Сорокина	w
74	9 Юлия	w	5 Сорокина	w
75	5 Алена	w	12 Татаренкова	w
76	12 Анастасия	w	12 Татаренкова	w
77	2 Светлана	w	12 Татаренкова	w
78	11 София	w	12 Татаренкова	w
79	9 Юлия	w	12 Татаренкова	w
80	5 Алена	w	11 Турыгина	w
81	12 Анастасия	w	11 Турыгина	w
82	2 Светлана	w	11 Турыгина	w
83	11 София	w	11 Турыгина	w
84	9 Юлия	w	11 Турыгина	w
85	5 Алена	w	9 Фёдорова	w
86	12 Анастасия	w	9 Фёдорова	w
87	2 Светлана	w	9 Фёдорова	w
88	11 София	w	9 Фёдорова	w
89	9 Юлия	w	9 Фёдорова	w

7. В базе данных есть две таблицы для хранения объявлений: **advs** и **closed_advs**, в первой хранятся все объявления, а во второй только неактивные.

Раз в день объявления из таблицы **advs** копируются в **closed_advs** — некоторые объявления уже скопированы.

Скопируйте оставшиеся объявления в таблицу **closed_advs**.

```
INSERT INTO closed_advs (id, user_id, category_id, text, date)
SELECT id, user_id, category_id, text, date
FROM advs
WHERE advs.id NOT IN (SELECT id FROM closed_advs) AND closed;

SELECT * FROM closed_advs;
```

	id	user_id	category_id	text	date
1	3	675	14	Продам мебель на кухню	2018-02-14 23:32:43
2	7	1343	55	Куплю шубу норковую	2018-02-14 23:31:11
3	10	874	6	Продам ноутбук Sony	2018-02-15 01:45:05
4	11	145	12	Продам детское кресло	2018-02-15 06:00:04

8. База данных автосалона содержит 4 таблицы: **marks**, **models**, **cars** и **cached_cars**. В **marks** и **models** хранятся данные о марках и моделях автомобилей, а в **cars** сами автомобили.

Для ускорения доступа к данным в базу добавили промежуточную таблицу **cached_cars**, которая хранит уже готовые строки в виде «**марка модель, цвет**», а также текущую цену автомобиля.

Заполните таблицу **cached_cars** свежими данными. Обратите внимание, что некоторые данные в таблице **cached_cars** уже устарели и их нужно обновить.

```
TRUNCATE cached_cars;
```

```
INSERT INTO cached_cars
SELECT cars.id, CONCAT(marks.name, ' ', models.name, ', ', cars.color), cars.price
FROM cars JOIN models ON cars.model_id = models.id
      JOIN marks ON models.mark_id = marks.id;
SELECT * FROM cached_cars;
```

	🔑 id ↕	🚗 car ↕	💰 price ↕
1	1	Lada Kalina, баклажан	180000
2	2	Toyota Prius, зеленый	780000
3	3	Peugeot 307, черный	230000
4	4	Ford Fiesta, серебристый	450000
5	5	Ford Mondeo, зеленый	670000
6	6	Mazda Demio, желтый	220000
7	7	Toyota Land Cruiser, черный	2400000
8	8	Ford Focus, зеленый	369000
9	9	Nissan Juke, синий	560000
10	10	Ford Focus, черный	610000
11	11	BMW X5, желтый	3400000
12	12	Audi TT, серебристый	780000
13	13	Ford Kuga, красный	478000
14	14	Ford Mondeo, желтый	350000
15	15	Nissan Patrol, белый	2790000