

Министерство образования Республики Беларусь  
Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования  
Кафедра инженерной психологии и эргономики

Базы данных  
ОТЧЁТ  
по лабораторной работе №6  
«Транзакции»

Выполнил  
гр.310901

Студент:  
Усов А. М.

Проверила

Воробей А. В.

Минск 2025

## Транзакция 1. Лабораторная №4:

begin;

```
select * from user_ u where u.produceruserid is null;
```

```
select * from dream d where d.price < 25;
```

```
select * from "role";
```

```
select * from payment p where p.orderid = 'c219a71c-a74f-4396-bda5-84e0e7b5c521';
```

```
select * from review r where userid = '0b6ad028-8582-432b-af34-e83f090760a4';
```

```
select * from dreamcategory d ;
```

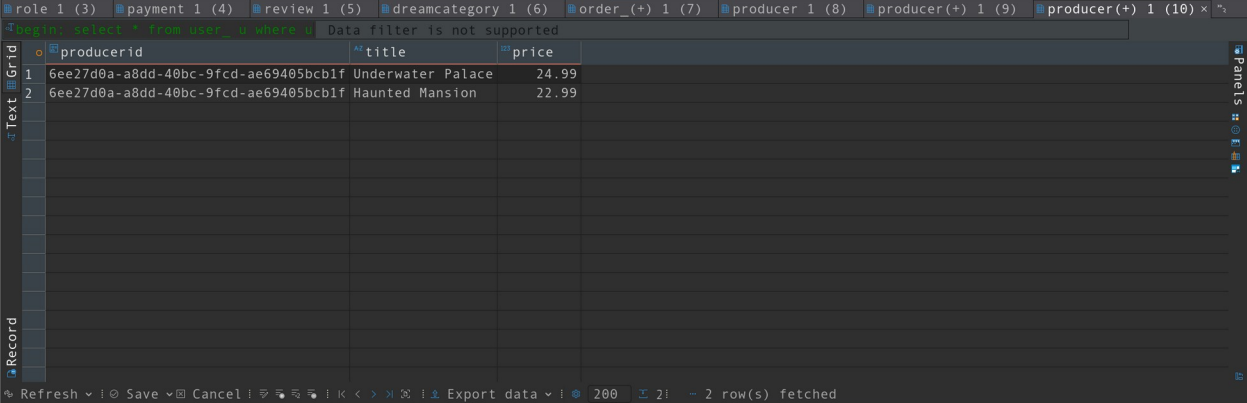
```
select
    o.userid ,
    d.title ,
    d.price ,
    xref.count
from order_ o
join orderdreamxref xref on o.orderid = xref.orderId
join dream d on xref.dreamid = d.dreamid
where o.orderid = '56002ce6-b61b-44b1-bf8e-6456a173d076';
```

```
select * from producer p ;
```

```
select p.title ,pu.secretkey , u."name"
from produceruser pu
join producer p on p.producerid = pu.producerid
join user_ u on pu.produceruserid = u.produceruserid
where p.title = 'Lucid Labs';
```

```
select p.producerid, d.title, d.price from dream d
join producer p on d.producerid = p.producerid
where p.title = 'Lucid Labs';
```

commit;



The screenshot shows a database management interface with a query window at the top and a results grid below. The query window contains the SQL statement: `begin; select * from user_ u where u Data filter is not supported`. The results grid displays two rows of data:

	producerid	title	price
1	6ee27d0a-a8dd-40bc-9fcd-ae69405bcb1f	Underwater Palace	24.99
2	6ee27d0a-a8dd-40bc-9fcd-ae69405bcb1f	Haunted Mansion	22.99

The interface includes a sidebar with 'Text' and 'Grid' views, and a bottom status bar indicating '200' and '2 row(s) fetched'.

## Транзакция 2. Лабораторная №5:

begin;

```
SELECT
    p.Title AS producer_title,
    COUNT(d.dreamid) as dream_count
FROM Producer p
JOIN Dream d ON p.ProducerId = d.ProducerId
GROUP BY p.Title
HAVING COUNT(d.DreamId) >= (
    SELECT AVG(dream_count)
    FROM (
        SELECT COUNT(DreamId) AS dream_count
        FROM Dream
        GROUP BY ProducerId
    ) AS counts
);
```

```
SELECT
    u.Name,
    AVG(r.Rating) AS avg_rating
FROM User_ u
JOIN Review r ON u.ConsumerUserId = r.UserId
GROUP BY u.ConsumerUserId, u.Name
HAVING AVG(r.Rating) > 4.0
ORDER BY avg_rating DESC;
```

```
SELECT ConsumerUserId, Name FROM User_
EXCEPT
SELECT DISTINCT u.ConsumerUserId, u.Name
FROM User_ u
JOIN Order_ o ON u.ConsumerUserId = o.UserId;
```

```
SELECT d.DreamId, d.Title
FROM Dream d
JOIN OrderDreamXRef odx ON d.DreamId = odx.DreamId
```

INTERSECT

```
SELECT DreamId, Title
FROM Dream;
```

```
WITH RankedDreams AS (
    SELECT
        d.Title AS dream_title,
        dc.Title AS category,
        d.Price,
        ROW_NUMBER() OVER (PARTITION BY dc.DreamCategoryId ORDER BY d.Price DESC) AS rn
```

```

FROM Dream d
JOIN DreamCategoryDreamXRef dcdx ON d.DreamId = dcdx.DreamId
JOIN DreamCategory dc ON dcdx.DreamCategoryId = dc.DreamCategoryId
)
SELECT dream_title, category, Price
FROM RankedDreams
WHERE rn <= 3;

```

```

SELECT
    p.Title AS producer,
    SUM(d.Price * odx.Count) AS total_revenue
FROM Producer p
JOIN Dream d ON p.ProducerId = d.ProducerId
JOIN OrderDreamXRef odx ON d.DreamId = odx.DreamId
JOIN Order_ o ON odx.OrderId = o.OrderId
GROUP BY p.ProducerId, p.Title
ORDER BY total_revenue DESC;

```

```

-- Пользователи, оставившие отзывы на BCE сны от 'DreamWorks Studio'
SELECT u.Name
FROM User_ u
WHERE NOT EXISTS (
    -- Найти сон от DreamWorks Studio, на который у пользователя НЕТ отзыва
    SELECT d.DreamId
    FROM Dream d
    JOIN Producer p ON d.ProducerId = p.ProducerId
    WHERE p.Title = 'DreamWorks Studio'
    EXCEPT
    SELECT r.DreamId
    FROM Review r
    WHERE r.UserId = u.ConsumerUserId
);

```

```

SELECT 'Dream Average Rating' AS metric, AVG(Dream.price) AS value
FROM Dream
UNION ALL
SELECT 'Review Average Rating', AVG(Rating)
FROM Review;

```

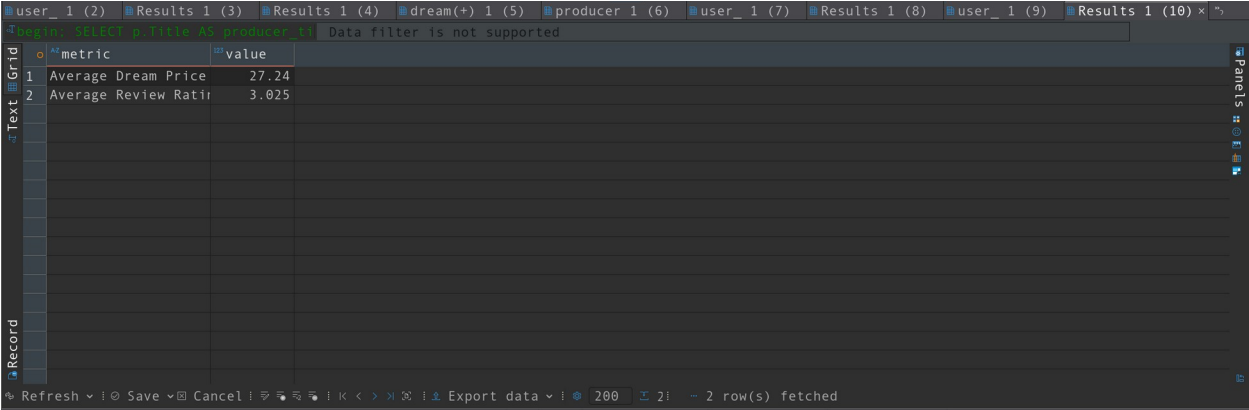
```

SELECT u.Name, u.Email
FROM User_ u
WHERE EXISTS (
    SELECT 1 FROM Order_ o WHERE o.UserId = u.ConsumerUserId
)
AND NOT EXISTS (
    SELECT 1 FROM Review r WHERE r.UserId = u.ConsumerUserId
);

```

```
SELECT 'Average Dream Price' AS metric, AVG(Price) AS value
FROM Dream
UNION ALL
SELECT 'Average Review Rating', AVG(Rating)
FROM Review;
```

commit;



The screenshot shows a database query results window with a table containing 2 rows and 2 columns. The columns are labeled 'metric' and 'value'. The first row shows 'Average Dream Price' with a value of 27.24. The second row shows 'Average Review Rating' with a value of 3.025. The window also displays a SQL query and a status bar indicating 2 rows were fetched.

	metric	value
1	Average Dream Price	27.24
2	Average Review Rating	3.025