```sql
-- Session 1

use sakila;

show variables like 'transaction_isolation';

-- read committed

set session transaction isolation level read committed;
start transaction;

select * from film where film_id = 15;

update film
set title = 'read committed 5'
where film_id = 15;

commit;

 -- repeatable read
set session transaction isolation level repeatable read;


start transaction;

select sum(price * quantity) as revenue from orders; -- 11624

commit;


-- table
-- 5 rows

-- as soon as you do first read : snapshot

-- transaction_id


start transaction;

select * from orders;
```

```sql
select sum(price * quantity) as revenue from orders;

update orders
set price = 40
where order_id = 12;

update orders
set price = 45
where price = 500;

commit;

set session transaction isolation level serializable;

start transaction;

select * from orders where order_id = 8;

insert into orders (order_id , price, quantity)
values (6 , 45 , 10);


update orders
set price = 15
where order_id = 7;

-- Error Code: 1213. Deadlock found when trying to get lock; try restarting
transaction


commit;

delete from orders
where order_id = 6;

SELECT
    t.PROCESSLIST_ID AS thread_id,
    t.PROCESSLIST_INFO AS query,
    l.OBJECT_SCHEMA AS locked_schema,
    l.OBJECT_NAME AS locked_table,
    l.INDEX_NAME AS locked_index,
    l.LOCK_TYPE AS lock_type,
    l.LOCK_MODE AS lock_mode,
```

```
    l.LOCK_DATA AS locked_data
FROM
    performance_schema.data_locks l
JOIN
    performance_schema.threads t ON l.THREAD_ID = t.THREAD_ID
WHERE
    l.OBJECT_SCHEMA = 'sakila' AND
    l.OBJECT_NAME = 'orders';
```

```
-- Session 2

use sakila;

show variables like 'transaction_isolation';

-- read committed
set session transaction isolation level read committed;
start transaction;

select * from film where film_id = 15;

update film
set title = 'read committed 4'
where film_id = 15;

commit;

-- repeatable read
set session transaction isolation level repeatable read;

start transaction;
```

```sql
 select sum(price * quantity) as revenue from orders;

 update orders
 set price = 15
 where order_id = 1;

commit;


start transaction;

select sum(price * quantity) as revenue from orders;

insert into orders (order_id , price, quantity)
values (8 , 500 , 10);

commit;


set session transaction isolation level serializable;

start transaction;

select * from orders where order_id IN (3 ,4 ,5 , 6 , 7);

select * from orders where order_id = 7;

select * from orders
where price = 45;

insert into orders (order_id , price, quantity)
values (13 , 500 , 10);

update orders
set price = 15
where order_id = 8;



commit;
```