

```
USE AdventureWorks;  
GO
```

```
--example #1: a blocked process  
--try to query AddressID = 1 with the default isolation level  
SELECT  
    AddressID  
    ,AddressLine1  
    ,AddressLine2  
    ,City  
    ,StateProvinceId  
    ,PostalCode  
    ,ModifiedDate  
FROM  
    Person.Address  
WHERE  
    AddressID = 1;
```

```
--what about getting the count?  
SELECT  
    COUNT(*)  
FROM  
    Person.Address
```

```
--try again to query AddressID = 1 with the default isolation level  
SELECT  
    AddressID  
    ,AddressLine1  
    ,AddressLine2  
    ,City  
    ,StateProvinceId  
    ,PostalCode  
    ,ModifiedDate  
FROM  
    Person.Address  
WHERE  
    AddressID = 1;
```

```
--(rollback the other query)
```

--example #3: lock compatibility

BEGIN TRANSACTION

--try to lock the resource with a shared lock

SELECT

 AddressID
 ,AddressLine1
 ,AddressLine2
 ,City
 ,StateProvinceId
 ,PostalCode
 ,ModifiedDate

FROM

 Person.Address WITH (HOLDLOCK)
 WHERE AddressID = 5;

--view the locks

SELECT

 resource_type
 ,(CASE
 WHEN resource_type = 'OBJECT' THEN
object_name(resource_associated_entity_id)
 WHEN resource_type IN ('DATABASE', 'FILE', 'METADATA') THEN 'N/A'
 WHEN resource_type IN ('KEY', 'PAGE', 'RID') THEN (
 SELECT
 object_name(object_id)
 FROM
 sys.partitions
 WHERE
 hobt_id=resource_associated_entity_id)
 ELSE
 'Undefined'
 END) AS resource_name
 ,request_mode as lock_type
 ,request_description
 ,request_status
 ,request_session_id
 ,request_owner_id AS transaction_id

FROM

 sys.dm_tran_locks

```
WHERE
    resource_type <> 'DATABASE';

ROLLBACK TRANSACTION;

--try to lock the resource with an exclusive lock
BEGIN TRANSACTION
    SELECT
        AddressID
        ,AddressLine1
        ,AddressLine2
        ,City
        ,StateProvinceId
        ,PostalCode
        ,ModifiedDate
    FROM
        Person.Address WITH (XLOCK)
        WHERE AddressID = 5;

ROLLBACK TRANSACTION;

--(query 1)

--example 5: lock granularity
--dirty reads with nolock
SELECT
    AddressID
    ,AddressLine1
    ,AddressLine2
    ,City
    ,StateProvinceId
    ,PostalCode
    ,ModifiedDate
FROM
    Person.Address WITH (NOLOCK)
WHERE
    AddressID = 1;

--example 6: dirty reads with read uncommitted
```

```
--change the isolation level
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
GO
--read the uncommitted row
SELECT
    AddressID
    ,AddressLine1
    ,AddressLine2
    ,City
    ,StateProvinceId
    ,PostalCode
    ,ModifiedDate
FROM
    Person.Address
WHERE
    AddressID = 1;

--(query 2)

--example 7: setting the timeout property
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
GO

SET LOCK_TIMEOUT 5000;
GO

--query the locked resource
SELECT
    AddressID
    ,AddressLine1
    ,AddressLine2
    ,City
    ,StateProvinceId
    ,PostalCode
    ,ModifiedDate
FROM
    Person.Address
WHERE
    AddressID = 1;

--return to the default behavior
```

```
SET LOCK_TIMEOUT -1;
```

```
GO
```

```
--another approach is to use the READPAST hint
```

```
--query the locked resource
```

```
SELECT
```

```
    AddressID
  ,AddressLine1
  ,AddressLine2
  ,City
  ,StateProvinceId
  ,PostalCode
  ,ModifiedDate
```

```
FROM
```

```
    Person.Address WITH (READPAST)
```

```
WHERE
```

```
    AddressID = 1;
```

```
--(query 1)
```

```
--example #8: using snapshot isolation
```

```
--set the connection's isolation level to the default
```

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
GO
```

```
--attempt to read data
```

```
SELECT
```

```
    AddressID
  ,AddressLine1
  ,AddressLine2
  ,City
  ,StateProvinceId
  ,PostalCode
  ,ModifiedDate
```

```
FROM
```

```
    Person.Address;
```

```
--set the connection's isolation level to the snapshot
```

```
SET TRANSACTION ISOLATION LEVEL SNAPSHOT;
```

```
GO
```

```
--attempt to read data
SELECT
    AddressID
    ,AddressLine1
    ,AddressLine2
    ,City
    ,StateProvinceId
    ,PostalCode
    ,ModifiedDate
FROM
    Person.Address;

--query 1
```