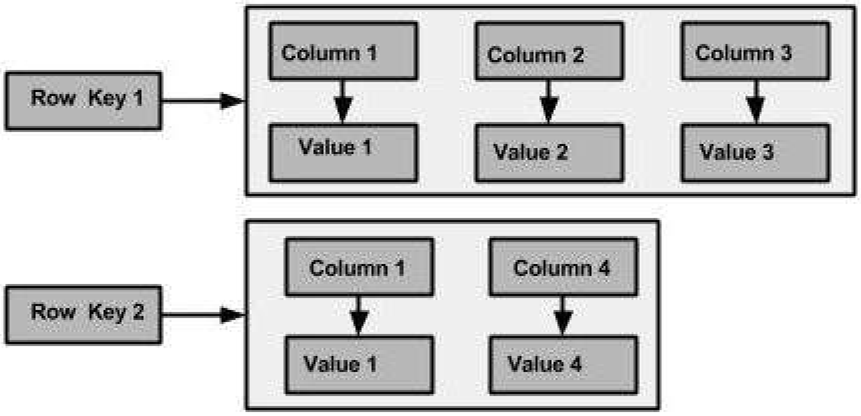


- Cassandra is a key-value database
- WHERE clause generally needs to include Partition key

```
describe keyspaces;
use user_transaction;
```

The following query works based on the partition key

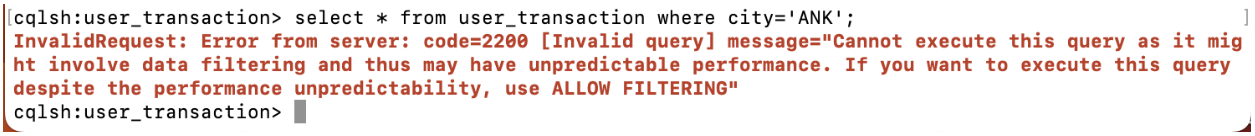
```
select * from user_transaction where person_id = 1;
```

The following query also works. Because we initially defined `person_id, transaction_id` as a partition key

```
select * from user_transaction where person_id = 1 and transaction_id >
1;
```

The following query wouldn't work, because we haven't defined city as a partition key, and city is not index

```
select * from user_transaction where city='ANK';
```

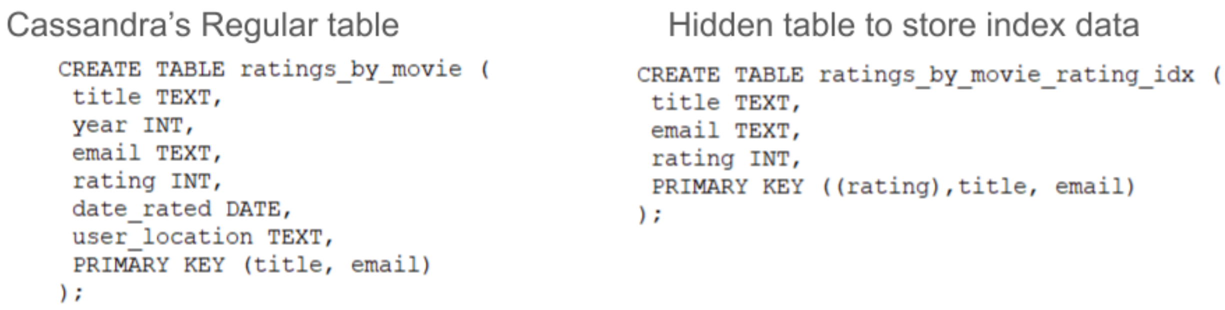


```
cqlsh:user_transaction> select * from user_transaction where city='ANK';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it mig
ht involve data filtering and thus may have unpredictable performance. If you want to execute this query
despite the performance unpredictability, use ALLOW FILTERING"
cqlsh:user_transaction>
```

# SECONDARY INDEX

With Secondary Index, columns can be used via **WHERE** query.

Cassandra creates a hidden table for the secondary index on each node.



Cassandra's Regular table

```
CREATE TABLE ratings_by_movie (
 title TEXT,
 year INT,
 email TEXT,
 rating INT,
 date_rated DATE,
 user_location TEXT,
 PRIMARY KEY (title, email)
);
```

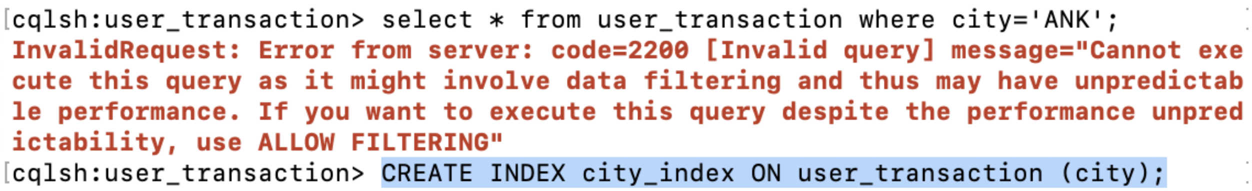
Hidden table to store index data

```
CREATE TABLE ratings_by_movie_rating_idx (
 title TEXT,
 email TEXT,
 rating INT,
 PRIMARY KEY ((rating),title, email)
);
```

```
Creating a secondary index doesn't increase performance. It allows to
create a query in the WHERE clause.
```

### How to create a secondary index?

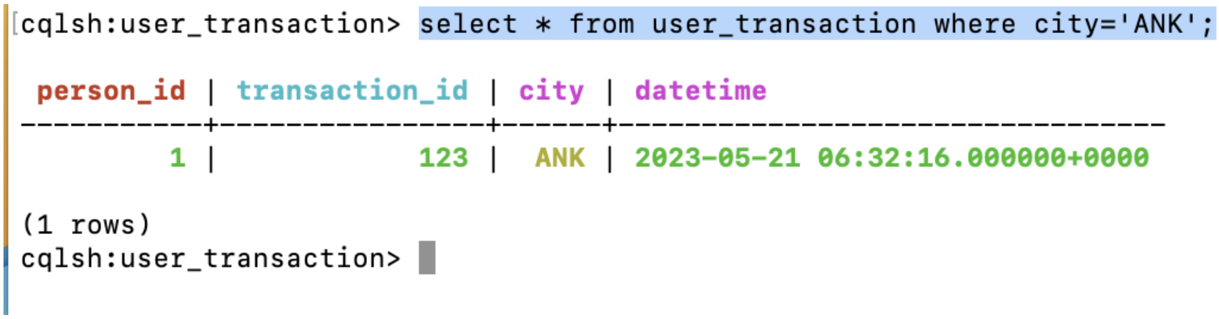
```
CREATE INDEX city_index ON user_transaction (city);
```



```
cqlsh:user_transaction> select * from user_transaction where city='ANK';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot exe
cute this query as it might involve data filtering and thus may have unpredictab
le performance. If you want to execute this query despite the performance unpred
ictability, use ALLOW FILTERING"
cqlsh:user_transaction> CREATE INDEX city_index ON user_transaction (city);
```

Execute SELECT query again

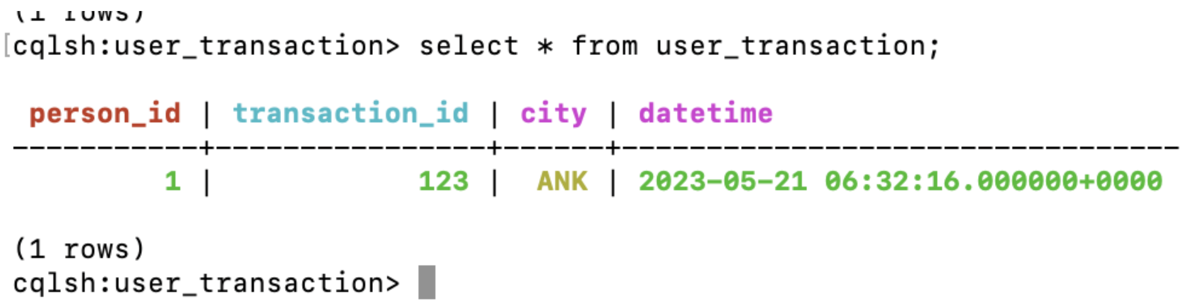
```
select * from user_transaction where city='ANK';
```



```
cqlsh:user_transaction> select * from user_transaction where city='ANK';

 person_id | transaction_id | city | datetime
-----------+----------------+------+---------------------------------
         1 |            123 |  ANK | 2023-05-21 06:32:16.000000+0000

(1 rows)
cqlsh:user_transaction>
```

# INCREASE THE SPEED OF THE QUERY

```
Creating a secondary index doesn't increase performance. It allows to
create a query in the WHERE clause.

To increase the speed of the query, you need to create a table specifically
for the table.
```



```
(1 rows)
cqlsh:user_transaction> select * from user_transaction;

 person_id | transaction_id | city | datetime
-----------+----------------+------+---------------------------------
         1 |            123 |  ANK | 2023-05-21 06:32:16.000000+0000

(1 rows)
cqlsh:user_transaction>
```

You are responsible for RUD operations

```
CREATE TABLE CITY_AND_TRANSACTION_ID ( transaction_id bigint,  city
text, PRIMARY KEY(transaction_id, city));
```