IST769 Homework Submission Template

Basic Information

Your Name: Mark Roberts  
Your SUID: 598273961  
Your Email: mrober04@syr.edu  
Date Due: September 7, 2021   
Homework #: 9

Your Answers:

1. Design your own scenario for which a Cassandra table would be a good solution. Make sure to explain the scenario and the specific characteristics of the scenario which would make Cassandra a good fit. Make sure to follow a query first approach and justify how the partition and cluster keys should be setup.

|  |
| --- |
| CODE |
| CREATE KEYSPACE stox with replication = {'class': 'SimpleStrategy', 'replication\_factor': '3'};  describe KEYSPACES;  use stox;  CREATE KEYSPACE spotify with replication = {'class': 'SimpleStrategy', 'replication\_factor': '3'};  describe KEYSPACES;  use spotify; |
| SCREENSHOT/OUTPUT |
|  |
| A scenario I thought of is Music Collection > Playlist. Collection would be Artists, Albums, Songs. Playlist would cull any number of them under a collection. I build the table and fed data. I thought next step could be to develop listen tally counts, additions, deletions. |

1. Create your Cassandra table in CQL based on your scenario from the previous exercise. You should define the columns and data types to suit your scenario in addition to configuring the partition and cluster keys.

|  |
| --- |
| CODE |
| CREATE TABLE stocks (symbol text, company\_name text, PRIMARY KEY(symbol));  describe table stocks;  INSERT INTO stocks (symbol, company\_name) VALUES ('AAPL', 'Apple');  INSERT INTO stocks (symbol, company\_name) VALUES ('NVDA', 'Nvidia');  INSERT INTO stocks (symbol, company\_name) VALUES ('AMZN', 'Amazon');  INSERT INTO stocks (symbol, company\_name) VALUES ('EMPH', 'Enphase');  INSERT INTO stocks (symbol, company\_name) VALUES ('DIS', 'Disney');  INSERT INTO stocks (symbol, company\_name) VALUES ('MSFT', 'Microsoft');  INSERT INTO stocks (symbol, company\_name) VALUES ('AMD', 'AMD');  INSERT INTO stocks (symbol, company\_name) VALUES ('JPM', 'JP Morgan Chase');  INSERT INTO stocks (symbol, company\_name) VALUES ('CSCO', 'Cisco');  SELECT \* FROM stocks;  CREATE TABLE playlist (artist text, song text, PRIMARY KEY(artist));  describe table playlist;  INSERT INTO playlist (artist, song) VALUES ('Throwing Muses', 'Sunray Venus');  INSERT INTO playlist (artist, song) VALUES ('Radiohead', 'Paranoid Android');  INSERT INTO playlist (artist, song) VALUES ('Tegan and Sara', 'Closer');  INSERT INTO playlist (artist, song) VALUES ('Bettie Serveert', 'Brain Tag');  INSERT INTO playlist (artist, song) VALUES ('Sleater-Kinney', 'No Cities to Love');  INSERT INTO playlist (artist, song) VALUES ('Belly', 'Feed the Tree');  SELECT \* FROM playlist; |
| SCREENSHOT/OUTPUT |
|  |

1. Write CQL statements to add data to your table. Add at least 9 records consisting of 3 different partition and cluster keys

|  |
| --- |
| CODE |
| CREATE TABLE stock\_quotes (exchange text, symbol text, offered\_on timestamp, bid\_price float, ask\_price float, PRIMARY KEY (symbol, offered on));  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:00', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:05', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:10', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:15', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:20', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:25', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:30', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:35', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:40', 250.00, 251.00);  INSERT INTO stock\_quotes (exchange, symbol, offered\_on, bid\_price, ask\_price) VALUES ('NSDQ', 'AAPL', '2019-11-01 10:00', 250.00, 251.00);  SELECT \* FROM stock\_quotes WHERE bid\_price>250;  SELECT \* FROM stock\_quotes WHERE bid\_price>250 ALLOW FILTERING; |
| SCREENSHOT/OUTPUT |
|  |

1. Write a CQL statement to create an index or materialized view on your table so that you can set a different partition key to prevent ALLOW FILTERING. Then write a CQL SELECT statement to demonstrate it works as designed.

|  |
| --- |
| CODE |
| CREATE INDEX ix\_stock\_quotes on stox.stock\_quotes (bid\_price);  SELECT \* FROM stock\_quotes WHERE bid\_price>250;  SELECT \* FROM stock\_quotes WHERE bid\_price=250;  SELECT \* FROM stock\_quotes WHERE ask\_price>250; |
| SCREENSHOT/OUTPUT |
|  |

1. Write a CQL statement to create an index or materialized view on your table so that you can set a different cluster key to prevent ALLOW FILTERING. Then write a CQL SELECT statement to demonstrate it works as designed.

|  |
| --- |
| CODE |
| DROP MATERIALIZED VIEW IF EXISTS stock\_quotes\_by\_ask\_price;  CREATE MATERIALIZED VIEW stock\_quotes\_by\_ask\_price AS SELECT \* FROM stock\_quotes WHERE symbol IS NOT NULL AND offered\_on IS NOT NULL AND ask\_price IS NOT NULL PRIMARY KEY (symbol, offered\_on, ask\_price);  SELECT \* FROM stock\_quotes\_by\_ask\_price WHERE symbol='AAPL' AND offered\_on >= '2019-11-01'; |
| SCREENSHOT/OUTPUT |
|  |