1. Creating Users

```
SELECT current_user();
SHOW USERS;
CREATE OR REPLACE USER MMALHOTRA2;
```

2. Key_Pair Authentication

```
ALTER USER MMALHOTRA SET RSA_PUBLIC_KEY='xyz';

DESC USER MMALHOTRA;
```

3. Creating Stock_Data Schema

```
--DDL Statement  - Schema for FORX DATA TABLE
CREATE or replace TABLE STOCK_DATA(
   id INTEGER IDENTITY(1,1)
  ,Volume INTEGER
  ,Volume_weighted_avg_pric float
  ,Opening_Price float
  ,Closing_Price float
  ,Highest_Price float
  ,Lowest_Price  float
  ,No_of_Transactions INTEGER
  ,Start_Timestamp timestamp
  ,Ticker   VARCHAR(16777216)
  ,load_timestamp timestamp default TO_TIMESTAMP_NTZ(current_timestamp));



  --Select Statement
select * from FOREX_DATA
limit 5;
```

4. Creating Forex_Data_Schema

```
create or replace database TRADEDATA;

use TRADEDATA;


--DDL Statement  - Schema for FORX DATA TABLE
CREATE or replace TABLE FOREX_DATA(
   id INTEGER IDENTITY(1,1)
  ,Volume INTEGER
  ,Volume_weighted_avg_pric float
```

```
   ,Opening_Price float
   ,Closing_Price float
   ,Highest_Price float
   ,Lowest_Price  float
   ,No_of_Transactions INTEGER
   ,Start_Timestamp timestamp
   ,Ticker   VARCHAR(16777216)
   ,load_timestamp timestamp default TO_TIMESTAMP_NTZ(current_timestamp));



   --Select Statement
select * from FOREX_DATA
limit 5;


CREATE or replace TABLE FOREX_DATA_RESTORE(
    id INTEGER IDENTITY(1,1)
   ,Volume INTEGER
   ,Volume_weighted_avg_pric float
   ,Opening_Price float
   ,Closing_Price float
   ,Highest_Price float
   ,Lowest_Price  float
   ,No_of_Transactions INTEGER
   ,Start_Timestamp timestamp
   ,Ticker   VARCHAR(16777216)
   ,load_timestamp timestamp default TO_TIMESTAMP_NTZ(current_timestamp));
```

5. Creating view to see correlation between different currencies:

```
USE DATABASE TRADEDATA;


CREATE VIEW correlation_matrix_view AS
SELECT
  a.Ticker AS ticker_a,
  b.Ticker AS ticker_b,
  CORR(a.Closing_Price, b.Closing_Price) AS correlation
FROM
  forex_data a
JOIN
  forex_data b
ON
```

```
    a.Start_Timestamp = b.Start_Timestamp
    AND a.Ticker < b.Ticker
  GROUP BY
    a.Ticker, b.Ticker;
```

6. Setting User Privileges

ASSIGN ROLE FOR
1. ANALYST: VIEW ACCESS 2. DEVELOPER : VIEW AND EDIT

```
DROP VIEW daily_summary_view;

CREATE VIEW daily_summary_view AS
SELECT
  TICKER,
  DATE_TRUNC('day', START_TIMESTAMP) AS trading_date,
  CLOSING_PRICE,
  AVG(CLOSING_PRICE) OVER (PARTITION BY TICKER ORDER BY
START_TIMESTAMP ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) AS
FIVE_DAY_ROLLING_AVG
FROM
  FOREX_DATA;
```

7. Some Backup Operations

```
CREATE STAGE TRADE_DATA_BACKUP;

DROP FILE FORMAT forex_backup_file_format;

CREATE FILE FORMAT forex_backup_file_format
  TYPE = CSV
  FIELD_DELIMITER = ','
  SKIP_HEADER = 1
  NULL_IF = ('')
  FIELD_OPTIONALLY_ENCLOSED_BY = '"';


CREATE OR REPLACE TASK my_backup_task
  WAREHOUSE = TRADE_DATA_WH
  SCHEDULE = 'USING CRON 0 0 * * * UTC'-- run the task daily
  AS
  COPY INTO @TRADE_DATA_BACKUP
  FROM FOREX_DATA
  FILE_FORMAT = forex_backup_file_format;

SELECT * FROM SNOWFLAKE.ACCOUNT_USAGE.STAGES;

-- ALTER TASK my_backup_task RUN NOW;
```

```
SHOW TASKS IN TRADEDATA.public;

ALTER TASK my_backup_task RESUME;
-- RESUME TASK my_backup_task;

EXECUTE TASK my_backup_task;

USE DATABASE TRADE_DATA;

SELECT *
FROM @TRADE_DATA_BACKUP;

--restore from backup


COPY INTO FOREX_DATA_RESTORE
FROM (SELECT * FROM @TRADE_DATA_BACKUP/data_0_0_0.csv.gz)
FILE_FORMAT = (TYPE = CSV, COMPRESSION = 'gzip', SKIP_HEADER = 1)
ON_ERROR = CONTINUE;
```