SQL query code using ML Forecasting and Tasks

Query 1: Load Data into Snowflake

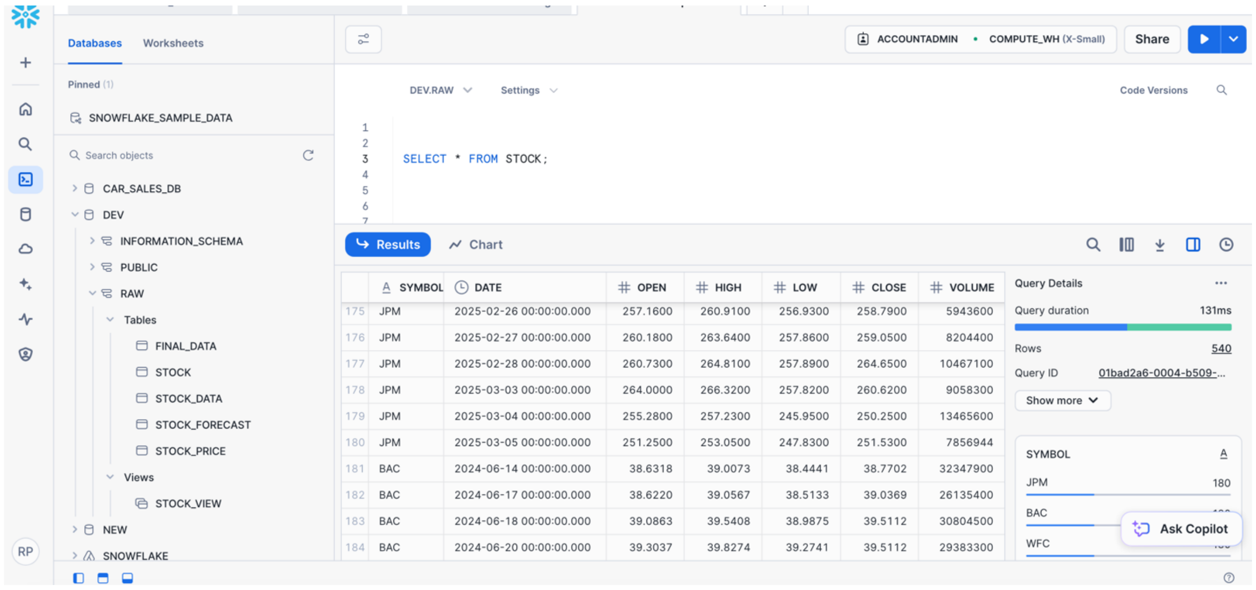
```
# Task to load data into Snowflake
@task
def load(records, target_table):
    con = return_snowflake_conn()

    try:
        con.execute("BEGIN;")
        con.execute(f"""CREATE TABLE IF NOT EXISTS
{target_table} (
            symbol string,
            date timestamp,
            open number(38, 4),
            high number(38, 4),
            low number(38, 4),
            close number(38, 4),
            volume number(38, 0),
            PRIMARY KEY (symbol, date)
          )""")

        con.execute(f"""DELETE FROM {target_table}""")

        for r in records:
            symbol = r[0]
            date = r[1]
            open_price = r[2]
            high = r[3]
            low = r[4]
            close = r[5]
            volume = r[6]

            sql = f"""
                    INSERT INTO {target_table} (symbol, date, open,
high, low, close, volume)
                    VALUES (%s, %s, %s, %s, %s, %s, %s)
                  """
            con.execute(sql, (symbol, date, open_price, high, low,
close, volume))
        con.execute("COMMIT;")
    except Exception as e:
        con.execute("ROLLBACK;")
        print(e)
        raise e
```



```sql
SELECT * FROM STOCK;
```

	SYMBOL	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME
175	JPM	2025-02-26 00:00:00.000	257.1600	260.9100	256.9300	258.7900	5943600
176	JPM	2025-02-27 00:00:00.000	260.1800	263.6400	257.8600	259.0500	8204400
177	JPM	2025-02-28 00:00:00.000	260.7300	264.8100	257.8900	264.6500	10467100
178	JPM	2025-03-03 00:00:00.000	264.0000	266.3200	257.8200	260.6200	9058300
179	JPM	2025-03-04 00:00:00.000	255.2800	257.2300	245.9500	250.2500	13465600
180	JPM	2025-03-05 00:00:00.000	251.2500	253.0500	247.8300	251.5300	7856944
181	BAC	2024-06-14 00:00:00.000	38.6318	39.0073	38.4441	38.7702	32347900
182	BAC	2024-06-17 00:00:00.000	38.6220	39.0567	38.5133	39.0369	26135400
183	BAC	2024-06-18 00:00:00.000	39.0863	39.5408	38.9875	39.5112	30804500
184	BAC	2024-06-20 00:00:00.000	39.3037	39.8274	39.2741	39.5112	29383300

Query Details
Query duration 131ms
Rows 540
Query ID 01bad2a6-0004-b509-...

SYMBOL	
JPM	180
BAC	
WFC	

Fig 1: Input Data Loaded to Snowflake

Query 2: ML forecasting task to train the model

```
# Task 1: ML forecasting task to train the model
@task
def train(train_input_table, train_view,
forecast_function_name):
    hook =
SnowflakeHook(snowflake_conn_id='snowflake_default')
    conn = hook.get_conn()
    cur = conn.cursor()

    create_view_sql = f"""
    CREATE OR REPLACE VIEW {train_view} AS
    SELECT
        CAST(DATE AS TIMESTAMP_NTZ) AS DATE,
CLOSE, SYMBOL
    FROM {train_input_table};
    """

    create_model_sql = f"""
    CREATE OR REPLACE SNOWFLAKE.ML.FORECAST
{forecast_function_name} (
        INPUT_DATA => SYSTEM$REFERENCE('VIEW',
'{train_view}'),
        SERIES_COLNAME => 'SYMBOL',
        TIMESTAMP_COLNAME => 'DATE',
        TARGET_COLNAME => 'CLOSE',
        CONFIG_OBJECT => {{ 'ON_ERROR': 'SKIP' }}
    );
    """

    try:
        cur.execute(create_view_sql)
        cur.execute(create_model_sql)
        cur.execute(f"CALL
{forecast_function_name}!SHOW_EVALUATION_METRICS
();")
    except Exception as e:
        print(e)
        raise
    finally:
        cur.close()
        conn.close()
```

Query 3: Generating predictions from the model

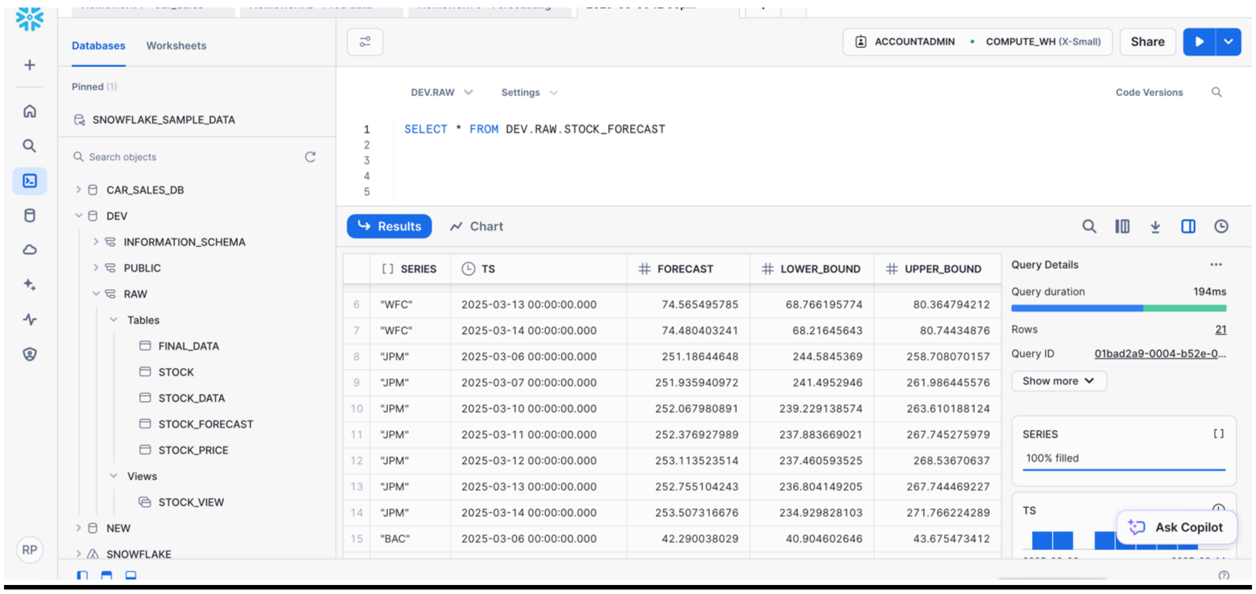
```
# Task 2: Generating predictions from the model
@task
def predict(forecast_function_name, train_input_table,
forecast_table, final_table):
    hook =
SnowflakeHook(snowflake_conn_id='snowflake_default')
    conn = hook.get_conn()
    cur = conn.cursor()

    make_prediction_sql = f"""
    BEGIN
        CALL {forecast_function_name}!FORECAST(
            FORECASTING_PERIODS => 7,
            CONFIG_OBJECT => {{'prediction_interval': 0.95}}
        );
        LET x := SQLID;
        CREATE OR REPLACE TABLE {forecast_table} AS
SELECT * FROM TABLE(RESULT_SCAN(:x));
    END;
    """

    create_final_table_sql = f"""
    CREATE TABLE IF NOT EXISTS {final_table} AS
    SELECT SYMBOL, DATE, CLOSE AS actual, NULL AS
forecast, NULL AS lower_bound, NULL AS upper_bound
    FROM {train_input_table}
    UNION ALL
    SELECT REPLACE(series, '"', '') AS SYMBOL, ts AS
DATE, NULL AS actual, forecast, lower_bound, upper_bound
    FROM {forecast_table};
    """

    try:
        cur.execute(make_prediction_sql)
        cur.execute(create_final_table_sql)
    except Exception as e:
        print(e)
        raise
    finally:
        cur.close()
        conn.close()
```

Output Screenshots

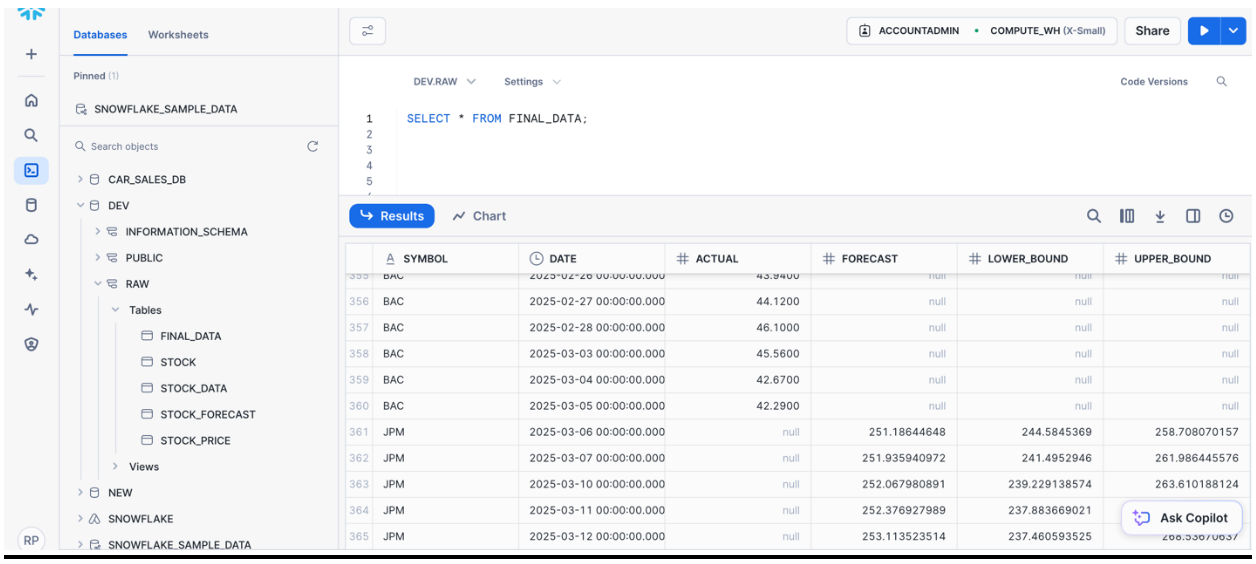


```sql
SELECT * FROM DEV.RAW.STOCK_FORECAST
```

	SERIES	TS	FORECAST	LOWER_BOUND	UPPER_BOUND
6	"WFC"	2025-03-13 00:00:00.000	74.565495785	68.766195774	80.364794212
7	"WFC"	2025-03-14 00:00:00.000	74.480403241	68.21645643	80.74434876
8	"JPM"	2025-03-06 00:00:00.000	251.18644648	244.5845369	258.708070157
9	"JPM"	2025-03-07 00:00:00.000	251.935940972	241.4952946	261.986445576
10	"JPM"	2025-03-10 00:00:00.000	252.067980891	239.229138574	263.610188124
11	"JPM"	2025-03-11 00:00:00.000	252.376927989	237.883669021	267.745275979
12	"JPM"	2025-03-12 00:00:00.000	253.113523514	237.460593525	268.53670637
13	"JPM"	2025-03-13 00:00:00.000	252.755104243	236.804149205	267.744469227
14	"JPM"	2025-03-14 00:00:00.000	253.507316676	234.929828103	271.766224289
15	"BAC"	2025-03-06 00:00:00.000	42.290038029	40.904602646	43.675473412

Query Details
Query duration 194ms
Rows 21
Query ID 01bad2a9-0004-b52e-0...
Show more
SERIES [ ]
100% filled
TS

Fig 2: Forecast Table



```sql
SELECT * FROM FINAL_DATA;
```

	SYMBOL	DATE	ACTUAL	FORECAST	LOWER_BOUND	UPPER_BOUND
355	BAC	2025-02-26 00:00:00.000	43.9400	null	null	null
356	BAC	2025-02-27 00:00:00.000	44.1200	null	null	null
357	BAC	2025-02-28 00:00:00.000	46.1000	null	null	null
358	BAC	2025-03-03 00:00:00.000	45.5600	null	null	null
359	BAC	2025-03-04 00:00:00.000	42.6700	null	null	null
360	BAC	2025-03-05 00:00:00.000	42.2900	null	null	null
361	JPM	2025-03-06 00:00:00.000	null	251.18644648	244.5845369	258.708070157
362	JPM	2025-03-07 00:00:00.000	null	251.935940972	241.4952946	261.986445576
363	JPM	2025-03-10 00:00:00.000	null	252.067980891	239.229138574	263.610188124
364	JPM	2025-03-11 00:00:00.000	null	252.376927989	237.883669021	[illegible]
365	JPM	2025-03-12 00:00:00.000	null	253.113523514	237.460593525	268.53670637

Fig 3: Actual vs Predicted Data