This is not mandatory reading, but here's the code we'll run in the "Stored Procedures" videos. It may come in handy when you're doing the associated hands-on assignment.

---> list all procedures

SHOW PROCEDURES;

SELECT \* FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER

LIMIT 100;

---> see the latest and earliest order timestamps so we can determine what we want to delete

SELECT MAX(ORDER\_TS), MIN(ORDER\_TS) FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER;

---> save the max timestamp

SET max\_ts = (SELECT MAX(ORDER\_TS) FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER);

SELECT $max\_ts;

SELECT DATEADD('DAY',-180,$max\_ts);

---> determine the necessary cutoff to go back 180 days

SET cutoff\_ts = (SELECT DATEADD('DAY',-180,$max\_ts));

---> note how you can use the cutoff\_ts variable in the WHERE clause

SELECT MAX(ORDER\_TS) FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER

WHERE ORDER\_TS < $cutoff\_ts;

USE DATABASE TASTY\_BYTES;

---> create your procedure

CREATE OR REPLACE PROCEDURE delete\_old()

RETURNS BOOLEAN

LANGUAGE SQL

AS

$$

DECLARE

max\_ts TIMESTAMP;

cutoff\_ts TIMESTAMP;

BEGIN

max\_ts := (SELECT MAX(ORDER\_TS) FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER);

cutoff\_ts := (SELECT DATEADD('DAY',-180,:max\_ts));

DELETE FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER

WHERE ORDER\_TS < :cutoff\_ts;

END;

$$

;

SHOW PROCEDURES;

---> see information about your procedure

DESCRIBE PROCEDURE delete\_old();

---> run your procedure

CALL DELETE\_OLD();

---> confirm that that made a difference

SELECT MIN(ORDER\_TS) FROM TASTY\_BYTES\_CLONE.RAW\_POS.ORDER\_HEADER;

---> it did! We deleted everything from before the cutoff timestamp

SELECT $cutoff\_ts;