-- Sales\_Order\_Detail

CREATE TABLE Sales\_Order\_Detail(

SalesID CHAR(10) NOT NULL,

ItemID CHAR(10) NOT NULL,

ItemName VARCHAR(50),

Unit VARCHAR(10),

Quantity INT,

Price DECIMAL(10, 2),

Amount DECIMAL(12, 2)

);

-- Insert values into table of “Sales\_Order\_Detail”

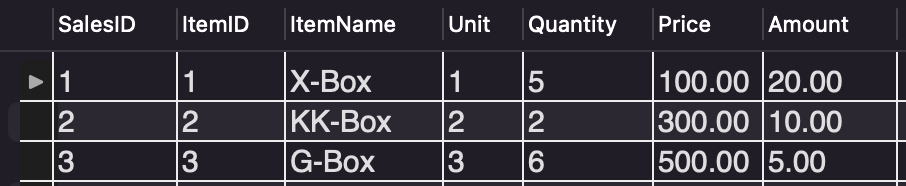
insert into Sales\_Order\_Detail (SalesID, ItemID, ItemName, Unit, Quantity, Price, Amount ) values (1, 1, 'X-Box', 1, 5, 100, 20);

insert into Sales\_Order\_Detail (SalesID, ItemID, ItemName, Unit, Quantity, Price, Amount ) values (2, 2, 'KK-Box', 2, 2, 300, 10);

insert into Sales\_Order\_Detail (SalesID, ItemID, ItemName, Unit, Quantity, Price, Amount ) values (3, 3, 'G-Box', 3, 6, 500, 5);

-- Query all rows of column from Sales\_Order\_Detail and present

Select \* from Sales\_Order\_Detail;

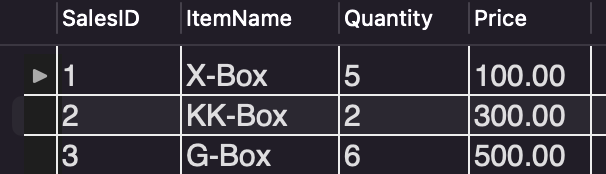


-- VIEW

Create view v\_Sales\_Order\_Detail AS select SalesID, ItemName, Quantity, Price from Sales\_Order\_Detail;

-- Query all rows of column from v\_Sales\_Order\_Detail and present

Select \* from v\_Sales\_Order\_Detail;



-- Update the row of column from v\_Sales\_Order\_Detail and present

Update v\_Sales\_Order\_Detail set SalesID = 100 where SalesID = 2;

select \* from v\_Sales\_Order\_Detail;

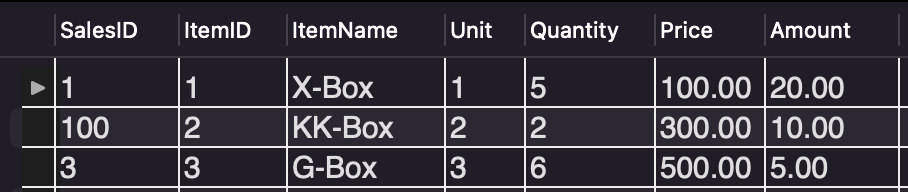
A screenshot of a computer

Description automatically generated with low confidence

-- Query all rows of column from Sales\_Order\_Detail and present

-- You will see the original table of value itself, being updated

select \* from Sales\_Order\_Detail;



-- A kindly Reminder:

/\*

The rows of column that developer does query the VIEW table (i.e., v\_Sales\_Order\_Detail), is completely from the original data of Table: Sales\_Order\_Detail.

Developer can add, query, update, and delete in the VIEW table.

DB Engineer creates VIEW table, is likely to limit the accessed privilege of column(s) and only present the columns in need for developer

\*/