**Name: Jianwei Su**

**Date:12/18/2022**

**Gadde: 98%**

# **Exercises**

## **Level 4: Table Syntax**

***Note: For all SQL homeworks, just paste your code under the question in the Word document. No images or proof of output are necessary. Please make your answers bold and blue.***

Reminder: For all of the questions below no screenshots are necessary. Just provide the answers in English or the code you used to do it.

### **4.1: TABLE MODIFICATION**

1. Create a table called TEST\_TRADES. It should contain the same columns as our TRADE\_DATA table.

\**for the next four questions please use the TEST\_TRADES table*\*

**CREATE TABLE dbo.test\_trades(**

**COB\_DATE INT(10) NOT NULL,**

**POSITION\_ID TEXT (50) NOT NULL,**

**CUSIP** **TEXT(100) NULL,**

**EMPLOYEE\_ID TEXT(10) NOT NULL,**

**FUND\_ID TEXT(10) NOT NULL,**

**QUANTITY TEXT(20) NULL,**

**NOTIONAL\_USD TEXT(20) NULL,**

**SCHEDULE TEXT(45) NULL,**

**MATURITY\_DATE TEXT(10) NULL)**

1. Add an index to the column on the columns you feel appropriate.**4.1.2 Asked for an index not a column.**

**ALTER TABLE `dbo`.`test\_trades`**

**ADD INDEX `Index\_01` (`TRADER\_ID`(10) ASC, `FUND\_ID`(10) ASC, `QUANTITY`(30) ASC) VISIBLE;**

1. Change the column titled “EMPLOYEE\_ID” to “TRADER\_ID”.

**ALTER TABLE DBO.test\_trades**

**CHANGE COLUMN EMPLOYEE\_ID TRADER\_ID TEXT NULL DEFAULT NULL**

1. INSERT 100 rows from TRADE\_DATA into this TEST table, sort by date so the latest dates come in first.

**INSERT INTO DBO.test\_trades(COB\_DATE,POSITION\_ID,CUSIP,TRADER\_ID,FUND\_ID,QUANTITY,NOTIONAL\_USD,SCHEDULE,MATURITY\_DATE)**

**SELECT COB\_DATE,POSITION\_ID,CUSIP,EMPLOYEE\_ID,FUND\_ID,QUANTITY,NOTIONAL\_USD,SCHEDULE,MATURITY\_DATE**

**FROM DBO.trade\_data\_hist**

**ORDER BY**

**COB\_DATE DESC**

**LIMIT 100**

1. Update all of the rows where TRADER\_ID = ‘T1’ to say ‘INTERNAL’ instead.

**UPDATE DBO.test\_trades**

**SET TRADER\_ID='INTERNAL'**

**WHERE TRADER\_ID IN ('T1')**

### **4.2: VIEWS**

1. What is a view and why might it be useful?

**A view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. View might be useful because it can hide complexity(remove unnecessary columns and focus in central location for the main logic), be used for security purposes because they provide encapsulation of the name of the table and simplify supporting legacy code.**

1. Create a view that would be useful on our dataset (not the one from the lecture).

**CREATE VIEW DBO.QUANTITY\_TRADES\_VIEW AS**

**SELECT**

**DBO.trade\_data\_hist.COB\_DATE AS COB\_DATE,**

**DBO.trade\_data\_hist.POSITION\_ID AS POSITION\_ID,**

**DBO.trade\_data\_hist.EMPLOYEE\_ID AS EMPLOYEE\_ID,**

**DBO.trade\_data\_hist.FUND\_ID AS FUND\_ID,**

**DBO.trade\_data\_hist.QUANTITY AS QUANTITY**

**FROM**

**DBO.trade\_data\_hist**

**WHERE DBO.trade\_data\_hist.QUANTITY <> ''**

1. Explain why your view will be helpful and who might be using it at the firm

**This view will be helpful to view the quantity of trades. It is very clear to view the quantity of trades with which fund\_id, trader and date. If someone wants to make a report about quantity of trades, this would be a very clear and clean for him.**

### **4.3: UNDERSTANDING OUR DATA**

1. Which table contains information about our clients?

**I think FUND\_INFO  contains information about our clients.**

1. Which table contains information about our employees?

**EMPLOYEE\_INFO contains information about our employees.**

1. What is the purpose of SECURITY\_INFO?

**The purpose of SECURITY\_INFO is to show the information about the securities we trade. With SECURITY\_INFO we can see more information about the securities we have traded such as what sector they are in and what country they originated from.**

1. What table might you add to enrich our data further? **4.3.4 Asked for a new table with a new purpose, not new columns**

**Traders might trade options, stocks and futures, so I might add a table about trading types. It would be used to show which trader has traded how many quantities of which type of securities in which period.**

### **4.4: ROW VS COLUMN**

1. Explain some key advantages of row based vs column based database and vice-versa.

**Advantages of Row Oriented Data stores:**

**Records in Row Oriented Data stores are easy to read and write. Row oriented data stores are best suited for online transaction system.**

**Disadvantages of Row Oriented Data stores:**

**Data is stored and retrieved one row at a time and hence could read unnecessary data if some of the data in a row are required. These are not efficient in performing operations applicable to the entire datasets and hence aggregation in row-oriented is an expensive job or operations. Typical compression mechanisms which provide less efficient result than what we achieve from column-oriented data stores.**

**Advantages of** **Column Oriented Data stores:**

**In this type of data stores, data are stored and retrieve in columns and hence it can only able to read only the relevant data if required. Column Oriented stores are best suited for online analytical processing. These are efficient in performing operations applicable to the entire dataset and hence enables aggregation over many rows and columns. These type of data stores basically permits high compression rates to little distinct or unique values in columns.**

**Disadvantages of Column Oriented Data stores:**

**In this type of data stores, read and write operations are slower as compared to row-oriented.**