**Name: Jianwei Su**

**Date:12/30/2022**

# **Exercises**

## **Level 6: Join & Union**

***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.***

Sample Question: How would you select all position ids from table TRADE\_DATA\_HIST?

A: **SELECT POSITION\_ID FROM DBO.TRADE\_DATA\_HIST**

### **6.1: INNER JOIN**

1. What is an inner join in your own words?

**Inner joins combine records from two tables whenever there are matching values in a field common to both tables.**

1. Using an inner join, select all of the columns from RISK\_DATA\_HIST but include the TICKER as well. Total columns returned should be ONLY and in this order: COB\_DATE, POSITION\_ID, TICKER, VAR

**SELECT TH.COB\_DATE,**

**TH.POSITION\_ID,**

**SI.TICKER,**

**RH.VaR**

**FROM trade\_data\_hist TH**

**INNER JOIN risk\_data\_hist RH**

**ON**

**TH.POSITION\_ID=RH.POSITION\_ID**

**INNER JOIN security\_info SI**

**ON TH.CUSIP=SI.CUSIP**

**AND TH.COB\_DATE BETWEEN SI.START\_DATE AND SI.END\_DATE**

1. Using multiple inner joins, please return the following columns, only for the first three days of available data: COB\_DATE, POSITION\_ID, TICKER, QUANTITY, VAR, SECTOR, COUNTRY

**SELECT TH.COB\_DATE,**

**TH.POSITION\_ID,**

**SI.TICKER,**

**TH.QUANTITY,**

**RH.VaR,**

**SI.SECTOR,**

**SI.COUNTRY**

**FROM trade\_data\_hist TH**

**INNER JOIN risk\_data\_hist RH**

**ON TH.POSITION\_ID=RH.POSITION\_ID**

**INNER JOIN security\_info SI**

**ON TH.CUSIP=SI.CUSIP**

**AND TH.COB\_DATE BETWEEN SI.START\_DATE AND SI.END\_DATE**

**WHERE TH.COB\_DATE <= 20180104**

### **6.2: LEFT/RIGHT JOIN**

1. Describe a left join in your own words.

**A left join returns all records from the left (first) table and the matched records from the right (second) table. If there is no match for a specific record, you'll get NULLs in the corresponding columns of the right table.**

1. Describe a right join.

**A right join** **returns all records from the right table (second), and the matching records from the left table (first). The result is 0 records from the left side, if there is no match.**

1. Write a logical left join using our dataset.

**SELECT RH.COB\_DATE,**

**RH.POSITION\_ID,**

**SI.TICKER,**

**RH.VaR**

**FROM risk\_data\_hist RH**

**LEFT JOIN security\_info SI**

**ON RH.CUSIP=SI.CUSIP**

**AND RH.COB\_DATE BETWEEN SI.START\_DATE AND SI.END\_DATE**

1. Write a logical right join using our dataset.

### **SELECT**

### **RH.COB\_DATE,**

### **RH.POSITION\_ID,**

### **SI.TICKER,**

### **RH.VaR**

### **FROM risk\_data\_hist RH**

### **RIGHT JOIN security\_info SI**

### **ON RH.CUSIP=SI.CUSIP**

### **AND RH.COB\_DATE BETWEEN SI.START\_DATE AND SI.END\_DATE**

### **ORDER BY**

### **COB\_DATE ASC**

### **6.3: FULL OUTER JOIN**

1. What is a full join?

**A FULL JOIN returns all the rows from both joined tables, whether they have a matching row or not.**

1. Why would you ever use a full join instead of an inner join?

**Inner join returns only the matching rows between both the tables, non-matching rows are eliminated. Full Join or Full Outer Join returns all rows from both the tables (left & right tables), including non-matching rows from both the tables.**

1. Fully join TRADE\_DATA\_HIST and RISK\_DATA\_HIST.

**since MySQL does not allow natural FULL JOINs, so I would do it via using pseudocodes**

**SELECT \* FROM trade\_data\_hist TH**

**FULL JOIN risk\_data\_hist RH**

**ON TH.POSITION\_ID=RH.POSITION\_ID**

1. Following question #3 above, do either of those tables contain positions not in the other?

**Yes they do!**

1. Now join your TEST\_DATA table on RISK\_DATA\_HIST (join only position\_ids). What did you learn from this join that’s different than the results of #3?

**SELECT \* FROM test\_trades TT**

**FULL JOIN risk\_data\_hist RH**

**ON TT.POSITION\_ID=RH.POSITION\_ID**

**Both of them contain the full table of RISK\_DATA\_HIST on the right side, but on the left side, TEST\_DATA table and TRADE\_DATA\_HIST table are different, the results show the difference between TEST\_DATA table and TRADE\_DATA\_HIST table on the left side. In short, TEST\_DATA table is a part of TRADE\_DATA\_HIST table, we can see #5’s result has more null values on the left side.**

### **6.4: SELF JOIN**

1. What is a self join?

**It is used to join a table to itself. This means that each row in a table is joined to itself and every other row in that table.**

1. List 3 examples (not from the lectures) of where you might use a self join?

**1. Daily/Monthly/Annual VaR changes on the same security**

**2. Daily/Monthly/Annual quantity changes on the same trader**

**3.** **Daily/Monthly/Annual convexity changes on the same security**

1. Which security on which date has the largest single day VaR increase in Jan 2018?

**SELECT**

**RH1.COB\_DATE,**

**RH1.CUSIP,**

**SUM(RH1.VaR) VaR1,**

**SUM(RH2.VaR) VaR2,**

**SUM(RH1.VaR)/SUM(RH2.VaR)-1 Single\_day\_VaR\_Changes**

**FROM risk\_data\_hist RH1**

**LEFT JOIN risk\_data\_hist RH2**

**ON RH1.CUSIP=RH2.CUSIP**

**AND RH1.COB\_DATE=DATE\_SUB(RH2.COB\_DATE, INTERVAL 1 DAY)**

**WHERE RH1.COB\_DATE<=20180131 AND RH1.CUSIP<>''**

**GROUP BY RH1.CUSIP**

**ORDER BY Single\_day\_VaR\_Changes DESC**

**Graphical user interface, text, application

Description automatically generated**

**Based on the results, Cusip=’C763405435’ on 20180111 has the largest single day VaR increase in Jan 2018.**

1. Which trader had the largest DECREASE in trade quantity between Jan 2 2018 and Feb 2 2018?

**SELECT**

**TH1.COB\_DATE,**

**TH1.EMPLOYEE\_ID,**

**SUM(TH1.QUANTITY) QTY1,**

**SUM(TH2.QUANTITY) QTY2,**

**SUM(TH1.QUANTITY)/SUM(TH2.QUANTITY) -1 QTY\_CHANGE**

**FROM trade\_data\_hist TH1**

**LEFT JOIN trade\_data\_hist TH2**

**ON TH1.EMPLOYEE\_ID=TH2.EMPLOYEE\_ID**

**AND TH1.COB\_DATE=DATE\_SUB(TH2.COB\_DATE, INTERVAL 1 DAY)**

**WHERE TH1.COB\_DATE BETWEEN 20180102 AND 20180202**

**GROUP BY TH1.COB\_DATE, TH1.EMPLOYEE\_ID**

**ORDER BY QTY\_CHANGE ASC**

Table

Description automatically generated

**Based on the results, T3 had the largest DECREASE in trade quantity between Jan 2 2018 and Feb 2 2018.**

### **6.5: UNION**

1. What is a Union?

**It is used to combine the data from the result of two or more SELECT command queries into a single distinct result set. This operator removes any duplicates present in the results being combined.**

1. How is a Union different than a Join?

|  |  |
| --- | --- |
| **Join:** | **Union:** |
| **JOIN combines data from many tables based on a matched condition between them** | **SQL combines the result-set of two or more SELECT statements.** |
| **It combines data into new columns.** | **It combines data into new rows.** |
| **Number of columns selected from each table may not be same.** | **Number of columns selected from each table should be same.** |
| **Datatypes of corresponding columns selected from each table can be different.** | **Datatypes of corresponding columns selected from each table should be same.** |
| **2It may not return distinct columns.** | **It returns distinct rows.** |

1. Provide the code to Union all columns from table TRADE\_DATA\_HIST and table RISK\_DATA\_HIST.

**Since SQL has strict rules for appending data: Both tables must have the same number of columns. TRADE\_DATA\_HIST has more columns than RISK\_DATA\_HIST, so I chose the same number of columns from TRADE\_DATA\_HIST to do the Union coding:**

**SELECT \***

**FROM risk\_data\_hist**

**UNION**

**SELECT**

**T.COB\_DATE,**

**T.POSITION\_ID,**

**T.CUSIP,**

**T.EMPLOYEE\_ID,**

**T.FUND\_ID,**

**T.QUANTITY**

**FROM trade\_data\_hist T**

1. Does the above Union make sense?

**No, it does not. Because it just stacked multiple datasets on top of each other.**

1. Union TEST\_DATA\_HIST onto TRADE\_DATA\_HIST. Explain why or why not this union makes sense.

**SELECT \* FROM trade\_data\_hist**

**UNION**

**SELECT \* FROM** **test\_trades**

**It does not make sense here. We know test\_trades has some of duplicate data from TRADE\_DATA\_HIST and it just stacked multiple datasets on top of each other without duplication, so it would just show as same as trade\_data\_hist.**

1. If you were going to Union two tables, one that has VaR on column 2 and one has Duration on column 2, what would happen to the output?

|  |  |
| --- | --- |
| **Column 1** | **Column 2** |
|  | **VaR1** |
|  | **VaR2** |
|  | **Duration1** |
|  | **Duration2** |

**Assuming each of two tables has only two rows, then it would show like above table after Union them.**

1. Do you have any suggestion how to deal with this assuming your manager insists you Union these two tables?

**We should create a VaR column and a Duration column and set the value to NULL where there is no data.**

1. What is a Union All?

**The UNION ALL command combines the result set of two or more SELECT statements (allows duplicate values).**

1. Provide a live case scenario where you might want to use Union All instead of Union.

**You would use UNION ALL when you really do need the multiple 'copies' of rows that would otherwise be removed when using UNION. For example, if you want to see how many multiple 'copies' of rows between tables, you should use Union All.**

1. Is there a way to do a Union All but not show certain duplicates? If yes, explain.

**Yes, it is.**

**Ex:**

**SELECT \* FROM mytable WHERE a=X UNION ALL SELECT \* FROM mytable WHERE b=Y AND a!=X**

### **6.6: Full Join Workaround**

1. Explain in English what we are trying to accomplish with this workaround

**Since MySQL does not support FULL OUTER JOIN, we can use left join, right join and union to achieve FULL OUTER JOIN in MySQL.**

1. Type out the code that is used in the video.

**SELECT TDH.POSITION\_ID, TDH.EMPLOYEE\_ID,RDH.POSITION\_ID,RDH.VAR**

**FROM trade\_data\_hist TDH LEFT JOIN risk\_data\_hist RDH**

**ON TDH.POSITION\_ID=RDH.POSITION\_ID**

**WHERE TDH.COB\_DATE BETWEEN 20180101 AND 20180201**

**UNION**

**SELECT TDH.POSITION\_ID, TDH.EMPLOYEE\_ID,RDH.POSITION\_ID,RDH.VAR**

**FROM trade\_data\_hist TDH RIGHT JOIN risk\_data\_hist RDH**

**ON TDH.POSITION\_ID=RDH.POSITION\_ID**

**WHERE TDH.COB\_DATE BETWEEN 20180101 AND 20180201**

**AND TDH.POSITION\_ID IS NULL**

1. Above, please add a comment ABOVE each line of code explaining what it does and why it is necessary to add.

**There are 10 lines of code above and here are the comment of each line:**

**1. Select columns we want to show from each table.**

**2.** **Do left join here: returns all records from the left (first) table and the matched records from the right (second) table .**

**3. Select columns do full join from each table.**

**4. Select certain time periods to avoid system crashing**

**5. Union: It is to remove the overlap from each table. That means it is used to remove all duplicates.**

**6. Select columns we want to show from each table.**

**7. Do right join here: returns all records from the right table (second), and the matching records from the left table (first).**

**8. Select columns do full join from each table.**

**9. Select certain time periods to avoid system crashing**

**10. It is supplement to the Union. It is not healthy to just rely on Union to remove the duplicates, so add it here.**

1. Write your own FULL JOIN WORKAROUND on TRADE\_DATA\_HIST and SECURITY\_PRICE\_HIST.

**SELECT TDH.COB\_DATE,TDH.CUSIP, TDH.EMPLOYEE\_ID,SPH.CLOSE\_PRICE\_USD, SPH.RATE**

**FROM trade\_data\_hist TDH LEFT JOIN security\_price\_hist SPH**

**ON TDH.CUSIP=SPH.CUSIP AND TDH.COB\_DATE=SPH.COB\_DATE**

**WHERE TDH.COB\_DATE BETWEEN 20180101 AND 20180201**

**UNION**

**SELECT TDH.COB\_DATE,TDH.CUSIP, TDH.EMPLOYEE\_ID,SPH.CLOSE\_PRICE\_USD, SPH.RATE**

**FROM trade\_data\_hist TDH RIGHT JOIN security\_price\_hist SPH**

**ON TDH.CUSIP=SPH.CUSIP AND TDH.COB\_DATE=SPH.COB\_DATE**

**WHERE TDH.COB\_DATE BETWEEN 20180101 AND 20180201**

**AND TDH.CUSIP IS NULL**

1. What did you learn from the output of this query?

**The output of this query returns all the rows of the columns I selected from both joined tables, whether they have a matching row or not.**