|  |  |  |  |
| --- | --- | --- | --- |
| Data Item | Location | Table | Comments |
| Stock Open/Close | SQL3/pvlchi6sql1, tickDB | Igtdev.dbo.tblStockHistory,  IvyDB.dbo.SECURITY\_PRICE,  DataMaster.dbo.tblEquityPrice, eventstats.stk\_clspx | >don’t use any other table besides Igtdev.dbo.tblStockHistory since those aren’t up to date (ex. tblStockHistory\_NEW)  >stk\_clspx has the security\_id tag to join to other tables  >be aware of how stocks are formatted! Ex. One table may have FAB.A as the stock whereas the other has FAB\A  >no info on how tables are constructed |
| Stock Live Prices | TickDB (git documentation is in peak6’s 7 github) | N/A | >creates a python dataframe. Syntax is “x = tickdb.read\_df(‘AA’, ‘2016-10-10’, start = “08:30:00”, end= “08:45:00”, security\_type = ‘option’, data\_type = ‘nbbo’, chunk = True)  >Unless you create a column for the stock symbol, default identifier is secid  >tickdb.refdata.symbol\_maping(sectype) needs something like “S:AA” as the input, can’t just be ‘AA’  >5 minute stock price can be pulled from ivydb.dbo.tblVwapStockPrice |
| Stock Volume | SQL3/pvlchi6sql1 | Igtdev.dbo.tblStockHistory |  |
| Option order volatility | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]), SQL3/pvlchi6sql1 | Events.fills | >gives you the nbbo/nbba vol  >IVs of all orders and their greeks can also be pulled from tickDB |
| Option prices | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]), SQL3/pvlchi6sql1, tickDB | Events (fills), ivyDB.dbo.OPTION\_PRICE, DataMaster.dbo.tblOptionPrice | >contains the nbbo/nbba prices of orders in the events and OPTION\_PRICE information  >tickDB best for getting option open/close prices. SQL and event capture tables only have best bid/offer from the day |
| Orders | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]) | Events (new\_######) | >Orders will have same root\_id but tables contain the different stages an order is in  >join based on root\_id with all other event capture queries  >will have stock name information under ‘underlying\_id’  >for parent level orders, specify “from\_pid = 0 and root\_id = order\_id) to grab them. Then join with the events.leg on root\_ids, pid = 0, to get the security\_id |
| Stock Orders | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]) | Events (xi\_######) | >Filter based on ‘StockXI’ since those will contain the stock orders  >join based on root\_id with all other event capture queries  >NOT ALL ORDERS WILL BE IN THE XI TABLE |
| Filled Orders | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]) | Events(fills) | >Will have information for Diego fills as well (check the crossing\_type column for this)  >order\_state doesn’t need to be “FILLED” in order to show a filled order. Partial fills will not have a “FILLED” order\_state for example  >join based on root\_id with all other event capture queries  >does have stock name information under  ‘security’ |
| Canceled orders | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]) | Events(canceled\_######, cancel\_####) | >The canceled\_###### will have information for option cancels  >The cancel\_##### will have information for stock cancels  >no information on the security name |
| Diego stocks | Event Capture in Postgres ( pg-exarc-uat:5500, pg-exarc-prd: 5500 [preferred]), tickDB | Events(fills), tickDB methods | >check for crossing\_type = ‘IGOOGI’/’INTERNALIZED’/etc to get the Diego stocks  >tickDB will get internal stock quotes and prints |
| PnL | SQL3/pvlchi6sql1 | Positions.dob.tblPnlPositions\_NG | >use SQL3 to get the most up to date on positions that WRT would have at the end of the day  >in SQL3, ‘ImpliedGamma’ is the Gamma that shows up on WRT  >need to join with Peak6Clearing.vwTradersActive to get the trader/trading group’s ID to join the name to the ID in the tblPnlPositions table  >filter by type = ‘P’. ‘T’ type is for trades that were just made so they won’t have greeks like Gamma\_C, thetaC, etc. |
| Computed Historical Vols | SQL3/pvlchi6sql1 | Igtdev.dbo.HistVoltbl | >contains weekly vols, but to have a better aligned timeframe window, you can pull weekly and weekly grouped daily vols from the DM “JG\_Weekly” |
| Rates | SQL3 | Companies.dbo.tblMPDDualRates, Peak6clearing.dbo.tblMPInterestRates (clearing firm interest rates, historical and current)  ,  Peak6Clearing.dbo.tblMPFOHardToBorrow (HTB rates, current and historical) | >last updated for the table was 12/16/2016  >table is manually entered, so need to pay attention to rate adjustment emails  >source is multiple clearing firms |
| Divs | SQL3/pvlchi6sql1 | Companies.dbo.tblDividendForcasts, ivydb.dbo.distribution (actual dividends) | >data gathered from Markit? |
| Reorgs/Distributions | SQL3/pvlchi6sql1 | Bloomberg.tblBBCorpActions, companies.dbo.tblSFBOptionCorpAction4\_Historical, Datamaster.dbo.tblAdjustedOptionPrice, DataMaster.tblAdjustedEquityPrice, companies.dbo.tblSFBDeliverable\_Historical | >tblSFBOptionCorpAction4\_Historical - adjustments to option position after corporate action  >tblAdjustedOptionPrice – adjust price of an option after corporate action  >tblAdjustedEquityPrice – adjust stock price of an option after corporate action  >tblSFBDeliverable\_Historical – multi hedged option changes for pricing and their deliverables when exercised |
| Events | SQL3/pvlchi6sql1, Event capture (pg-refdata-prd:5200) | Earnings (for event capture), companies.dbo.tblearnings, companies.tblPreAnnounce, companies.dbo.tblCalenderItems or companies.tblCalenderTypes (dates of announced events) |  |
| Trader IDs/names | SQL3/pvlchi6sql1 | Trades.dbo.tblTraders, peak6clearing.dbo.tblBusinessunits, peak6Clearing.dbo.tblBusinessUnitMemberDimension | >use the initials in tblTraders to join IDs with other tables. If table just uses the acronym, then use (CASE WHEN CHARINDEX('.', x.Account) = 0  then x.Account else LEFT(x.Account, CHARINDEX('.', x.Account) - 1) END) to strip out any “.VA” or other follow on acronyms |
| Auctions, books, flashes, spreads | SQL10 | tblAuctions | >contains PIP, flash, and book orders for both single options and spreads from various exchanges |
| Industries | SQL3/pvlchi6sql1 | Companies.dbo.tblIndustries, companies.tblCompanyGroups, companies.dbo.vwCompaniesbyGroup, IndexTrading.dbo.tblIndexComponents(ETFs and their weights) |  |
| Buy/Sell Limits or Robo Limits | SQL10 | Sparktools.tblRoboTraderLimits (5 minute snapshots), sparktools.tblBSLimitsUpdate (direct/raw limits) |  |
| Penny stock names | SQL3, Event capture (pg-exarc-prd:5500) | Companies.dbo.tblStocks, eventstats.penny\_symbol | >want any stock that ticks in pennies, go to public.penny\_list in event capture. Only issue is that that list was made in Dec 2016 so using it further out isn’t recommended |