

VoltDB Application Worksheet

Overview

- The purpose of this worksheet is to assist in the design of your VoltDB database schema. It is critical to understand both your data and your transactions to create an optimal design.
- Please read the VoltDB "Do's and Don'ts" document prior to filling out this worksheet.
- As an example, the worksheet includes a simple "Order Processing/Fulfillment" application in which orders are placed and held until delivery. After reviewing the example you can fill in your own worksheet for your application using the empty tables at the end of the document.

Step 1: Define your data model - Example

- list the tables for your application
- list the columns, including data type, and mark the potential partitioning keys
- include the projected maximum row count for each table and table growth rate information

Example - Order Processing Application

Table	Column	Partition Key	Data Type	Max # Rows	Growth Rate
order_header	order_id	?	integer	5,000,000	5% per year, orders are purged 1 week after fulfillment
	invoice_number	?	varchar(20)		
	customer_id	?	integer		
order_detail	order_id	?	integer	25,000,000	5% per year, orders are purged 1 week after fulfillment
	product_id		integer		
	order_quantity		integer		
	order_price		decimal		
product	product_id	?	integer	50,000	10% per year
	product_name		varchar(50)		
	product_price		decimal		
customer	customer_id	?	integer	1,000,000	10% per year
	customer_name		varchar(50)		

Step 2: Define your transactions - Example

- list the transactions for your application (both insert/update/delete and select types)
- include the projected frequency of each transaction (number of executions per time period)

Example - Order Processing Application

Transaction	Action	Table	Key	Frequency
Create/update an order	insert/update	order_header	order_id	50,000/sec
	insert/update	order_detail	order_id	
	select	product	product_id	
Lookup all orders for a particular customer	select	order_header	customer_id	25/sec
	select	order_detail	order_id	
	select	product	product_id	
Create/update a customer	insert/update	customer	customer_id	1/sec
Recreate an invoice	select	order_header	order_id	10,000/sec
	select	order_detail	order_id	
	select	customer	customer_id	
	select	product	product_id	
Create/update a product	insert/update	product	product_id	1/minute

Based on this workload, we should:

- Partition the **order_header** and **order_detail** tables, both on **order_id**. These tables are involved in the most frequent transactions and almost always accessed by this column (except for the "Lookup all orders for a particular customer").
- Replicate the **product** table since it is involved in the most frequent transactions (for reading) and its insert/update frequency is extremely low.
- Replicate the **customer** table because all rows are needed in all partitions for the "Recreate an invoice" transaction.

Step 2: Define your transactions - Your Application

- list the transactions for your application (both insert/update/delete and select types)
- include the projected frequency of each transaction (number of executions per time period)

Transaction	Action	Table	Key	Frequency

Based on this workload, I will:

- Partition table ?, on column ?.
- Partition table ?, on column ?.
- Replication table ?.