1. Introduction

1.1. Document purpose

The purpose of this document is to outline a data pipeline for overnight supplier loads in iDIG.

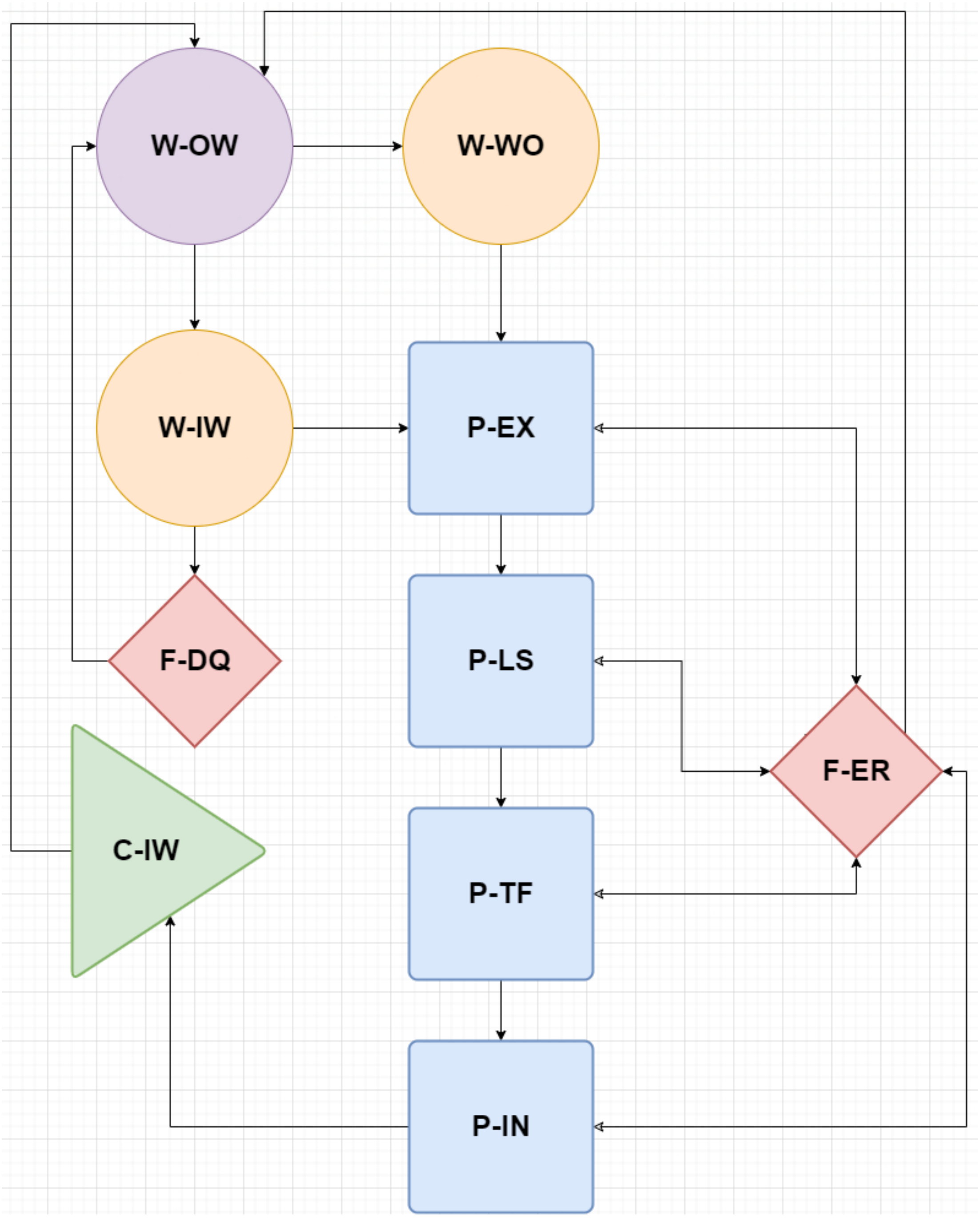
1.2. Introduction

VIP's iDIG solution is a web based application which allows clientele to create and save reports for analytical purposes. Clients can be either a distributor or a supplier in the beverage industry. For our intents and purposes, we will be focused on the latter.

Supplier reports rely on compiled data from their various distributors. The SRS team continually collects this distributor data and transforms it into supplier readable data. The iDIG data team then extracts this data, taking a snapshot of records at the time of load. It is the expectation of the supplier that data is accurate to the previous day. To meet this expectation, the iDIG data team loads data overnight. This window is, by default, 9pm to 9am EST, but can be customized to fit a specific supplier's needs. To complete this task, iDIG data maintains a linux service known as The Data Load Manager for Suppliers (DLMS).

The DLMS runs continuously from an ETL server, and at its most basic level, has two main tasks: To schedule nightly loads for all suppliers, and to place a wrapper around the iDIG data codebase to run each step of the ETL process. Within a load, there are several states which a supplier load can exist

The DLMS moves the user from between states based on the listed conditions. The DLMS reads the data from the DLMS configuration tables once every minute in order to determine which states must be updated. The default state is W-OW, where the customer sits until the next load window occurs.



1.3. Business requirements

- Allow AM to manually adjust the supplier load setting, such as start time, end time, or frequency.
- Allow AM to view the real time load summary to examine or analyze unusual data or status.
  - Checking the time record for each load to see if there is any unusual load time.

  - Checking the load step time average to see if there is any customer spending unusual time on one of the load steps.
  - Show all states of all customers.
- Analyze load and step data
- Display list of Supplier Information
- Display Which ETL server each queue will run on
- Display status of ETL CPU and RAM over time
  - This will help minimize slow or failing loads
- Display errors and records for each step
- Obtain suppliers current sales record count to determine which queue the supplier
- Reset and recover loads where needed
- Track states count by queue
- **Goal:** Keep supplier data up to date, this makes the client happy.

1.4. Definitions

- **DataChecker:** A VIP internal web application for iDIG and SRS support to monitor and manage customer loads, mainly for DLMS.
- **Data Load Manager for Suppliers:** An internal service for iDIG aimed at scheduling and running the supplier ETL process for iDIG.
- **Dequeued:** Load disqualification due to supplier missing load window.
- **Extract, Transform and Load:** A data integration process which takes data from multiple sources , and combines into a singular data warehouse.
- **ETL Servers/Boxes:** The Linux machines which run the DLMS service.
- **File Transfer Protocol:** The process of transferring files from server to client.
- **Full Load:** A DLMS load variant which extracts all data from the client’s current year as well as the previous 5 years. This load is kicked off manually by support in the following cases:
  - A supplier is out of balance.
  - SRS restates data upstream.
  - A new supplier is being initialized.
  - Upon request of support.
- **Held Load:** The supplier is placed in a hold state by request of the account manager. When a load is held, DLMS will skip the daily load until this flag is removed.

- **Incremental Load:** The default load type for DLMS loads. This load extracts sales records for only the previous three days, building on and updating the data in the supplier's current schema.
- **iDIG:** VIP's main reporting tool. This web application allows clients to create and save reports to track sales, inventory, route optimization, breakage and more.
- **iSeries:** An IBM transactional database operation system. DLMS uses this database as the starting point of the ETL.
- **KARMA:** A VIP survey tool downstream from iDIG's data warehouse.
- **Load Queue:** Supplier partitioning based on sales records.
- **Load Queue Minimum Sales Limit:** The minimum sales records required for a supplier to be allowed in the load queue.
- **Load Queue Overall Threshold:** The number of concurrent loads allowed by queue. This is inversely related to the load queue minimum sales limit.
- **Load Queue Step Threshold:** The number of concurrent loads allowed to run on the step.
- **Load Window:** The time frame for which a load may begin.
- **PyDIG:** A python package developed by the iDIG team to run load tasks.
- **Recover Step:** Restart from the beginning of the failed step.
- **Reset Load:** The failed load is discarded and the supplier is placed back to W-OW.
- **Snowflake:** A cloud based data warehouse service based in Montana
- **SnowDIG:** iDIG's supplier reporting warehouse hosted by Snowflake.
- **snowsql:** Snowflake's CLI tool for interacting with snowflake warehouses.
- **Supplier Reporting Services:** iDIG's upstream data provider. The SRS team compiles distributor data into a usable format for related suppliers.
- **The Extract Step:** The first step of a DLMS load. SRS data is extracted to the IFS in the supplier's specific folder. This data is commonly stored in .CSV or .DAT formats.
- **The Load Stage Step:** The second step of a DLMS load**.** This step is performed in the following order:
  - Compress IFS files to a gzip file format (.gz)
  - FTP compressed files to an ETL box
  - FTP file from ETL to stage on supplier schema using snowsql and Snowflake's **PUT** method.

  - Copy staged file to corresponding xtable using snowsql and Snowflake's **COPY INTO** method.
  - This is often done in parallel, loading up to 4 files simultaneously.
- **The Transform Step:** The third step of a DLMS load. This step transforms data in xtables to iDIG reporting tables and views.
- **The Indicators Step:** The fourth and final step of the DLMS load. After iDIG is loaded two files are loaded back to SRS from SnowDIG:
  - The Balance File: This shows the difference (if any) between SRS sales numbers and iDIG reporting data. Incremental loads can get out of balance over time, and this file provides a support metric to determine if a full load is needed.
  - The Indicator Extract: An extract from transformed sales data which is used by the KARMA team to update surveys.
- **The Integrated File System:** A file system on the iSeries in which extracted files are stored.
- **Vermont Information Processing (VIP):** Founded in 1972, VIP is a small Route Accounting software company based in Colchester, Vermont. VIP caters to the beverage industry in the United States and Canada, collecting and managing data for suppliers, distributors and retailers alike. VIP is 100% employee owned, and provides services for 1,900+ distributors and 1,100+ suppliers.
- **Waiting Outside Window:** The customer is outside their load window, and has completed their previous load successfully.
- **Waiting In Window:** The customer's load window has started, and their load may be started by the DLMS prior to the window closing.
- **Waiting With Override:** The customer has been queued to start a new load immediately. This is a manual override which ignores load windows.
- **xtable:** A table which is identical to a staged file, allowing file data to be queried during the transform step.

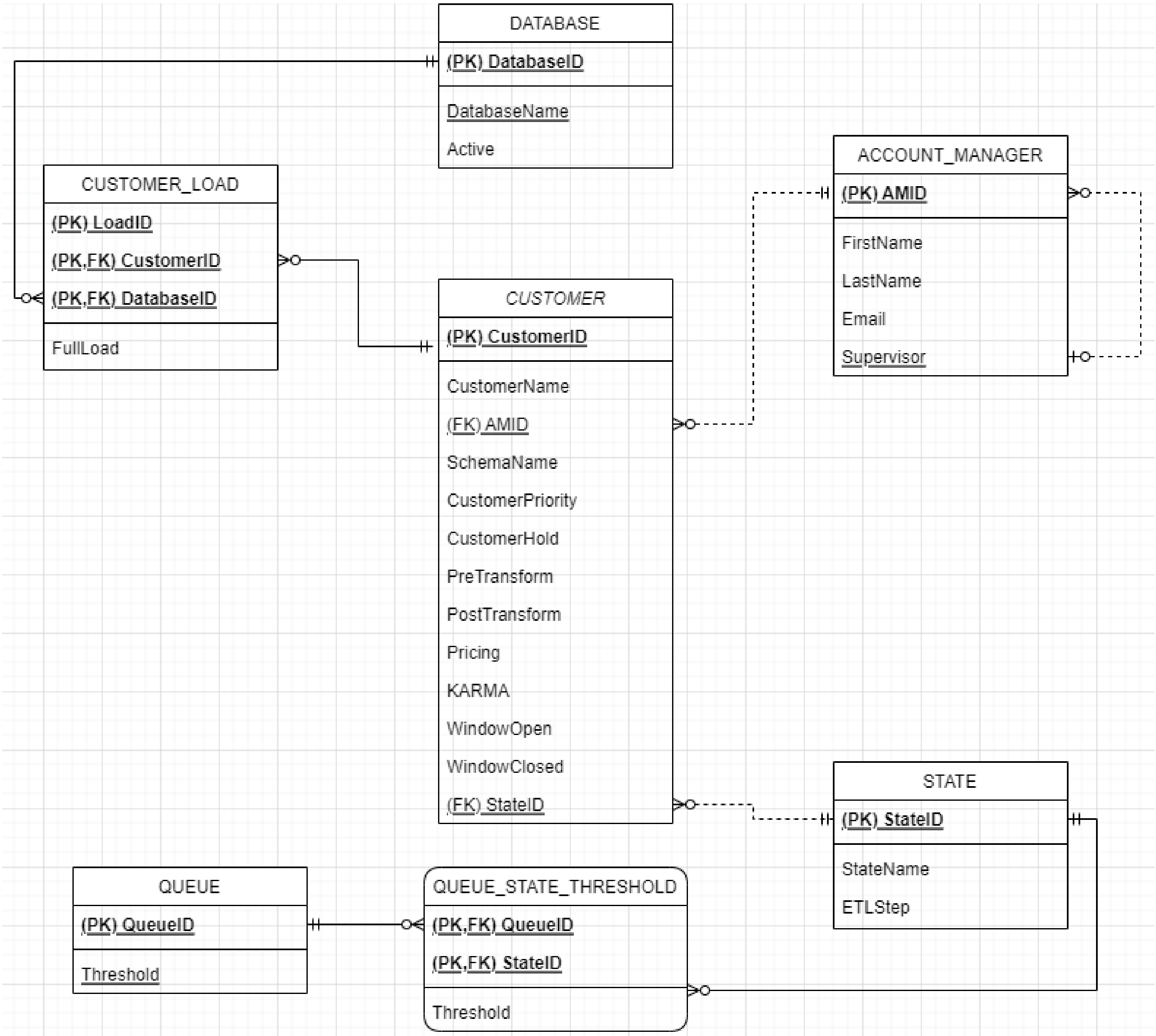
1.5. Acronyms and abbreviations

- **AM:** Account Manager
- **C-IW:** Completed in window.
- **DLMS:** Data Load Manager for Suppliers
- **ETL:** Extract, Transform and Load
- **F-DQ:** Dequeued
- **F-ER:** Errored
- **FTP:** File Transfer Protocol

- **FULL:** Full Load
- **INCR:** Incremental Load
- **IFS:** The Integrated File System
- **P-EX:** The Extract Step
- **P-IN:** The Indicator Step
- **P-LS:** The Load Stage Step
- **P-TF:** The Transform Step
- **SRS:** Supplier Reporting Services
- **VIP:** Vermont Information Processing
- **W-IW:** Waiting in window
- **W-OW:** Waiting outside window
- **W-WO:** Waiting with override

2. Data Model

2.1. E-R Data Model



2.2. Relationship table

Parent	*Child*	*Minimum*	*Maximum*
CUSTOMER	*CUSTOMER_LOAD*	*M-O*	*1-N*
CUSTOMER_LOAD	*DATABASE*	*M-O*	*1-N*
ACCOUNT_MANAGER	*CUSTOMER*	*M-O*	*1-N*
ACCOUNT_MANAGER	*ACCOUNT_MANAGER*	*O-O*	*0-N*
STATE	*CUSTOMER*	*M-O*	*1-N*
QUEUE	*QUEUE_STATE_THRESHOLD*	*M-O*	*1-N*
STATE	*QUEUE_STATE_THRESHOLD*	*M-O*	*1-N*

Database Design

Database: DLMS

Schemas: DLMS

Tables:

- ACCOUNT_MANAGER:

COLUMN	TYPE	REQUIRED	NULLABLE	DEFAULT	NOTES
AMID	INTEGER	yes	no	IDENTITY START 1 INCREMENT 1	PRIMARY KEY
FirstName	VARCHAR(12)	yes	no	‘’	
LastName	VARCHAR(20)	yes	no	‘’	
Email	VARCHAR(44)	no	no	lower(FirstName) || '.' || lower(LastName) || '@vtinfo.com'	
Supervisor	INTEGER	no	yes	NULL	This is a self referential FORIEGN KEY

- CUSTOMER

COLUMN	TYPE	REQUIRED	NULLABLE	DEFAULT	NOTES
AMID	INTEGER	yes	no	-1	FORIEGN KEY (ACCOUNT_MANAGER)
CustomerID	VARCHAR(3)	yes	no	‘’	PRIMARY KEY
CustomerName	VARCHAR(30)	yes	no	‘’	
CustomerPriority	INTEGER	no	no	5	<= 5
CustomerHold	BOOLEAN	no	no	FALSE	
KARMA	BOOLEAN	no	no	FALSE	
PostTransform	BOOLEAN	no	no	FALSE	
PreTransform	BOOLEAN	no	no	FALSE	
Pricing	BOOLEAN	no	no	FALSE	
SchemaName	VARCHAR(9)	no	no	'SRS' || CUSTOMER.ID || 'DIG'	This is overridden rarely, but can occur
StateID	VARCHAR(5)	yes	no	W-OW	FORIEGN KEY (STATE)
WindowClose	TIME(9)	yes	no	CAST('21:00:00' AS TIME(9))	Latest and auto load can start
WindowOpen	TIME(9)	yes	no	CAST('09:00:00' AS TIME(9))	Earliest auto load can start

- CUSTOMER_LOAD:

COLUMN	TYPE	REQ	NULL	DEFAULT	NOTES
CustomerID	VARCHAR(3)	yes	no	‘ ‘	FORIEGN KEY (CUSTOMER) PRIMARY KEY
DatabaseName	VARCHAR(8))	yes	no	IDIG_P01	FORIEGN KEY (DATABASE) PRIMARY KEY
LoadID	VARCHAR	yes	no	UUID_STRING()	PRIMARY KEY
FullLoad	BOOLEAN	yes	no	FALSE	

- DATABASE:

COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
DatabaseID	CHAR(3)	yes	no	“	PRIMARY KEY
DatabaseName	VARCHAR(8)	yes	no	‘ ‘	
Active	BOOLEAN	no	no	TRUE	

- QUEUE:

COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
QueueID	VARCHAR(20)	yes	no	‘ ‘	PRIMARY KEY
Threshold	INTERGER	no	yes	1000	The max concurrent loads within the given queue

- QUEUE_STATE_THRESHOLD:

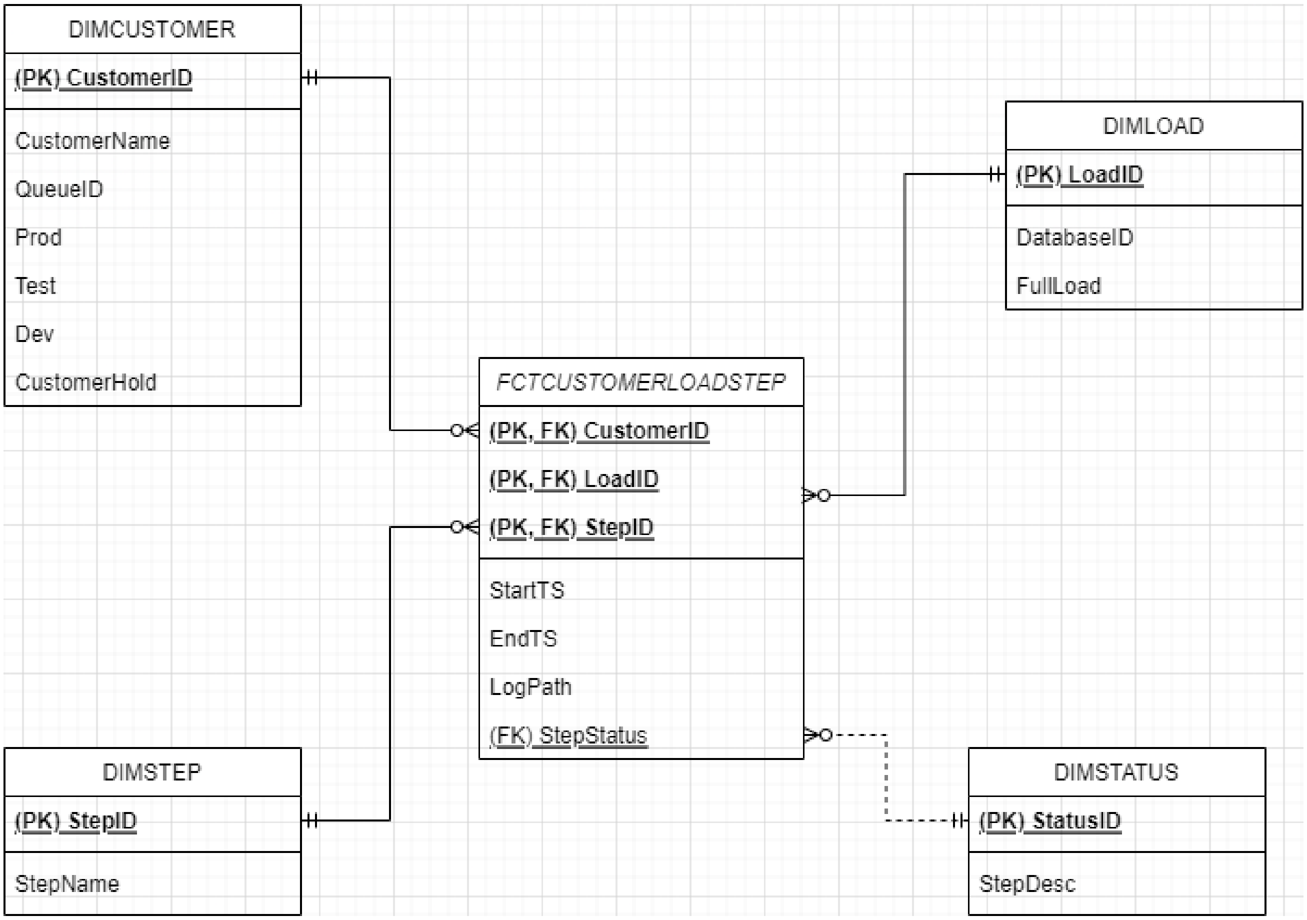
COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
QueueID	VARCHAR(20)	yes	no	‘ ‘	PRIMARY KEY, FORIEGN KEY (QUEUE)
StateID	VARCHAR(5)	yes	no	‘ ‘	PRIMARY KEY, FORIEGN KEY (STATE)
Threshold	INTEGER	no	yes	1000	The max concurrent instances of given step within a load

- STATE:

COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
StateID	VARCHAR(5)	yes	no	‘ ‘	PRIMARY_KEY
StateName	VARCHAR(30)	yes	no	‘ ‘	
ETLStep	BOOLEAN	no	no	FALSE	denotes if this state is a step within an etl load

Data Warehouse

Dimensional database design



Data Transformations

- DIMCUSTOMER
    - QueueID: Bin customers to Queue based on number of records in FCTSALES (found in production INFORMATION_SCHEMA.TABLES)
    - Prod: Check that SchemaName is in production database (IDIG_POC.INFORMATION_SCHEMA.TABLES)
    - Test: Check that SchemaName is in test database (IDIG_T01.INFORMATION_SCHEMA.TABLES)
    - Dev: Check that SchemaName is in development database (IDIG_D01.INFORMATION_SCHEMA.TABLES)
    - DLMS.CUSTOMER.CustomerID → CustomerID
    - DLMS.CUSTOMER.CustomerName → CustomerName
    - DLMS.CUSTOMER.CustomerHold → CustomerHold
- DIMLOAD

    - DLMS.CUSTOMER_LOAD.LoadID → LoadID
    - DLMS.CUSTOMER_LOAD.DatabaseID → DatabaseID
    - DLMS.CUSTOMER_LOAD.FullLoad → FullLoad
- DIMSTATUS: This table is unique to the warehouse, and only contains basic ID and description.
- DIMSTEP: Select StateID and StateName where ETLStep is True in DLMS.STATE, this becomes StepID and StepName Respectively

Data Warehouse Design

DIMCUSTOMER

COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
CustomerHold	BOOLEAN	yes	no	FASLE	
CustomerID	VARCHAR(3)	yes	no	‘ ‘	PRIMARY KEY
CustomerName	VARCHAR(30)	yes	no	‘ ‘	
Dev	BOOLEAN	no	no	FALSE	
Prod	BOOLEAN	no	no	TRUE	
Test	BOOLEAN	no	no	FALSE	
QueueID	VARCHAR(14)	yes	no	SNOWDIG_XSMALL	

DIMLOAD

COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
DatabaseID	VARCHAR(3)	no	no	P01	
FullLoad	BOOLEAN	no	no	FALSE	
LoadID	VARCHAR	yes	no	UUID_STRING()	PRIMARY KEY

DIMSTATUS

COLUMN	**TYPE**	**REQ**	**NULL**	**DEFAULT**	**NOTES**
StatusID	CHAR(1)	yes	no	‘’	PRIMARY KEY
StatusDesc	VARCHAR(30)	ye	no	‘’	

DIMSTEP

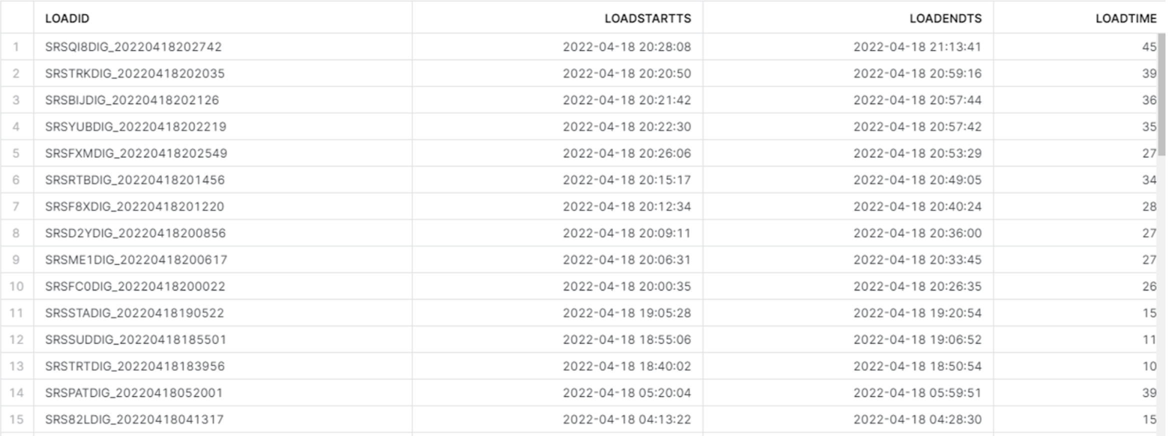
COLUMN	TYPE	REQ	NULL	DEFAULT	NOTES
StepID	VARCHAR(5)	yes	no	‘’	PRIMARY KEY
StepName	VARCHAR(30)	yes	no	‘’	

FCTLOADSTEP

<table>
<tr><th>COLUMN</th><th>TYPE</th><th>REQ</th><th>NULL</th><th>DEFAULT</th><th>NOTES</th></tr>
<tr><td>CustomerID</td><td>VARCHAR(3)</td><td>yes</td><td>no</td><td></td><td>PRIMARY KEY
FORIEGN KEY (DIMCUSTOMER)</td></tr>
<tr><td>EndTS</td><td>TIMESTAMP_NTZ(9)</td><td>no</td><td>yes</td><td></td><td></td></tr>
<tr><td>LoadID</td><td>VARCHAR</td><td>yes</td><td>no</td><td>UUID_STRING()</td><td>PRIMARY KEY
FORIEGN KEY (DIMLOAD)</td></tr>
<tr><td>LoadID</td><td>VARCHAR</td><td>no</td><td>yes</td><td>'$LOGDIR/' ||.CUSTOMERID || '/' || LOADID|| '/' || STEPID || '.log'</td><td></td></tr>
<tr><td>StartTS</td><td>TIMESTAMP_NTZ(9)</td><td>yes</td><td>no</td><td>CURRENT_TIMESTAMP()</td><td></td></tr>
<tr><td>StepID</td><td>VARCHAR(5)</td><td>yes</td><td>no</td><td></td><td>PRIMARY KEY
FORIEGN KEY (DIMSTEP)</td></tr>
<tr><td>StepStatus</td><td>VARCHAR(1)</td><td>yes</td><td>no</td><td>W</td><td>FORIEGN KEY (DIMSTATUS)</td></tr>
</table>

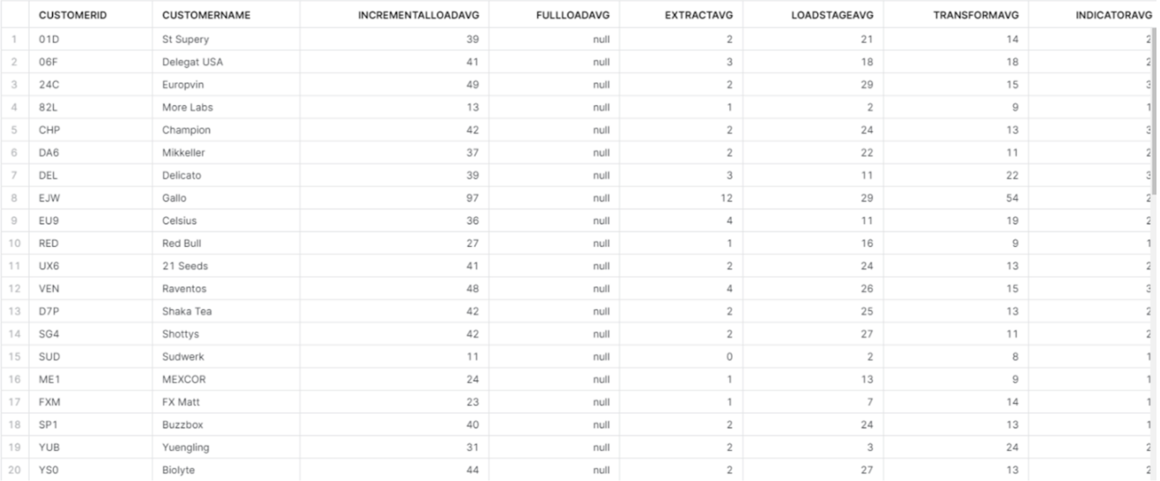
Views

1. LOAD_TIME: Gives start and end time for all completed/running load ids. Show the total time in minutes.

1. 

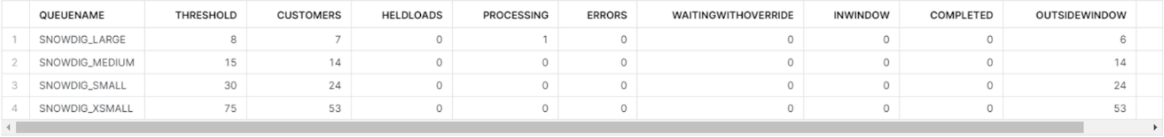
	LOADID	LOADSTARTTS	LOADENDTS	LOADTIME
1	SRSQI8DIG_20220418202742	2022-04-18 20:28:08	2022-04-18 21:13:41	45
2	SRSTRKDIG_20220418202035	2022-04-18 20:20:50	2022-04-18 20:59:16	39
3	SRSBIJDIG_20220418202126	2022-04-18 20:21:42	2022-04-18 20:57:44	36
4	SRSYUBDIG_20220418202219	2022-04-18 20:22:30	2022-04-18 20:57:42	35
5	SRSFXMDIG_20220418202549	2022-04-18 20:26:06	2022-04-18 20:53:29	27
6	SRSRTBDIG_20220418201456	2022-04-18 20:15:17	2022-04-18 20:49:05	34
7	SRSF8XDIG_20220418201220	2022-04-18 20:12:34	2022-04-18 20:40:24	28
8	SRSD2YDIG_20220418200856	2022-04-18 20:09:11	2022-04-18 20:36:00	27
9	SRSME1DIG_20220418200617	2022-04-18 20:06:31	2022-04-18 20:33:45	27
10	SRSFC0DIG_20220418200022	2022-04-18 20:00:35	2022-04-18 20:26:35	26
11	SRSSTADIG_20220418190522	2022-04-18 19:05:28	2022-04-18 19:20:54	15
12	SRSSUDDIG_20220418185501	2022-04-18 18:55:06	2022-04-18 19:06:52	11
13	SRSTRTDIG_20220418183956	2022-04-18 18:40:02	2022-04-18 18:50:54	10
14	SRSPATDIG_20220418052001	2022-04-18 05:20:04	2022-04-18 05:59:51	39
15	SRS82LDIG_20220418041317	2022-04-18 04:13:22	2022-04-18 04:28:30	15

2. AVERAGE_STEP_TIME: Shows average time for complete load and substeps by customer

    1. 

	CUSTOMERID	CUSTOMERNAME	INCREMENTALLOADAVG	FULLLOADAVG	EXTRACTAVG	LOADSTAGEAVG	TRANSFORMAVG	INDICATORAVG
1	01D	St Supery	39	null	2	21	14	2
2	06F	Delegat USA	41	null	3	18	18	2
3	24C	Europvin	49	null	2	29	15	3
4	82L	More Labs	13	null	1	2	9	1
5	CHP	Champion	42	null	2	24	13	3
6	DA6	Mikkeller	37	null	2	22	11	2
7	DEL	Delicato	39	null	3	11	22	3
8	EJW	Gallo	97	null	12	29	54	2
9	EU9	Celsius	36	null	4	11	19	2
10	RED	Red Bull	27	null	1	16	9	1
11	UX6	21 Seeds	41	null	2	24	13	2
12	VEN	Raventos	48	null	4	26	15	3
13	D7P	Shaka Tea	42	null	2	25	13	2
14	SG4	Shottys	42	null	2	27	11	2
15	SUD	Sudwerk	11	null	0	2	8	1
16	ME1	MEXCOR	24	null	1	13	9	1
17	FXM	FX Matt	23	null	1	7	14	1
18	SP1	Buzzbox	40	null	2	24	13	1
19	YUB	Yuengling	31	null	2	3	24	2
20	YS0	Biolyte	44	null	2	27	13	2

3. QUEUE_SUMMARY: Shows active states of all customers by queue.

    1. 

	QUEUENAME	THRESHOLD	CUSTOMERS	HELDLOADS	PROCESSING	ERRORS	WAITINGWITHOVERRIDE	INWINDOW	COMPLETED	OUTSIDEWINDOW
1	SNOWDIG_LARGE	8	7	0	1	0	0	0	0	6
2	SNOWDIG_MEDIUM	15	14	0	0	0	0	0	0	14
3	SNOWDIG_SMALL	30	24	0	0	0	0	0	0	24
4	SNOWDIG_XSMALL	75	53	0	0	0	0	0	0	53

    2. This is missing the Dequeue column as the screenshot would not allow all columns to show.

Since all of those are views, they will react to changes of their parent tables automatically.