Intelligent Process Data Warehouse for HPPM 5.0

Fabio Casati
Software Technology Laboratory
HP Laboratories Palo Alto

HPL-2002-120
April 26th , 2002*

E-mail: [casati] @hpl.hp.com

workflows, data warehousing, business intelligence

Business Process Management Systems log a large amount of operational data about processes and about the (human and automated) resources involved in their executions. This information can be analyzed for assessing the quality of business operations, identify problems, and suggest solutions. However, current process analysis systems lack the functionalities required to provide information that can be immediately digested and used by business analysts to take decisions. This paper presents a system and a set of techniques that overcome this limitations, enabling the use of log data for efficient business-level analysis of business processes.

* Internal Accession Date Only

Approved for External Publication

© Copyright Hewlett-Packard Company 2002

Intelligent Process Data Warehouse for HPPM 5.0

Fabio Casati
Hewlett-Packard Laboratories
1501 Page Mill Road, 1U-4
Palo Alto, CA, 94304 USA
[casati]@hpl.hp.com

Abstract

Business Process Management Systems log a large amount of operational data about processes and about the (human and automated) resources involved in their executions. This information can be analyzed for assessing the quality of business operations, identify problems, and suggest solutions. However, current process analysis systems lack the functionalities required to provide information that can be immediately digested and used by business analysts to take decisions. This paper presents a system and a set of techniques that overcome this limitations, enabling the use of log data for efficient business-level analysis of business processes.

OVERVIEW

HPPM 5.0 has the ability of logging a lot of information about the business processes it supports, including for instance the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event sent or received by a process.

This information is a gold mine for business and IT analysts: in fact, its analysis may *reveal problems* and inefficiencies in process executions and *identify solutions* in order to improve process execution *quality,* both as perceived by the users in terms of better and faster processes (*external* quality), and as perceived by service providers in terms of lower operating cost (*internal* quality). In addition, information on active processes (also collected by HPPM) can be used to *monitor* active process instances and be alerted of predicted and actual quality degradations.

The ***HPPM intelligent Process Data Warehouse*** (called Process Manager Warehouse, or PDW in the following) is a customizable, ready-to-go toolkit that allows analysts to quickly and easily extract performance and quality information from process execution logs, to monitor processes, detect problems, and identify solutions. PDW features include:

- ***Customized quality analysis.*** Users can analyze processes that exhibit *behaviors* of interest to the user, and identify the *causes* of such behaviors. A "behavior of interest" can be just about anything PDW users want to analyze, possibly because it corresponds to a particularly high- or low-quality execution. Examples of behaviors are process instances lasting more than 10 days, or in which John is involved, or in which node "re-send documents" is executed. PDW includes a large (and extensible) set of predefined behavior types.

- ***Semantic classification.*** Analysts can define *taxonomies* and instruct the PDW to classify process instances according to the defined taxonomies. For example, users can define classes "successful" and "unsuccessful" for a process and specify success criteria. PDW will then automatically classify process instances and show the results to the users. The analysis of process instances based on user-defined classification criteria turns out to be an extremely powerful tool for understanding and improving process execution quality.
- ***Pre-computed performance and quality metrics***. PDW contains a wide set of aggregated information describing typical performance metrics, such as the efficiency of a resource.
- ***Data verification and purification.*** PDW checks log data for consistency and corrects erroneous information that could make the analysis more complex and error-prone.
- ***Multidimensional analysis of process execution data*****.** PDW organizes data in a way that enables process analysis according to several *perspectives*, such as time, process initiators, resources, or services. For example, you can graphically see in how many process instances per week was John Doe involved during fiscal year 2001, or the average execution time of service "Travel reimbursement" during weekends.
- ***High-performance.*** Data are organized in order to maximize the performance of data analysis operations and deliver results quickly even when the analysis span across millions of process executions. Querying the PDW yields response times that are several hundreds times faster than querying the HPPM log tables.
- ***Preconfigured reports for the most common reporting tools.*** PDW includes configuration files for many commercial reporting tools, such as *Oracle Discoverer* or *Microsoft Excel*. Other reporting tools can also be used with minimal configuration effort, due to the richness of the PDW database that includes ready-to-use data. There is no need of writing complex and error-prone queries to get the information needed.
- ***Automatic data load.*** Data can be automatically extracted from HPPM and loaded into PDW. No user intervention is required. Manual loads are also possible.
- ***Monitoring and analysis of active instances.*** In addition to high-level and in-depth analysis of completed processes, PDW can also load and show data about active processes. It provides several ad-hoc views specifically designed to show data that is typically of interest for monitoring active processes, both from a technical and a business viewpoint.
- ***Ease of use.*** Despite the many user-defined features and the flexibility of the tool, the configuration is very simple, and complex analysis patterns can be specified by filling simple forms. No code is required to use the PDW, and the installation is straightforward.

PDW CONCEPTS

PDW Data Structure: *Facts* and *Dimensions*

We have been working with researcher, consultants, and customers to identify common customer needs and consequently design the PDW database to guarantee the best performance over a wide range of frequently asked queries and to simplify the definition of reports on top of PDW. The PDW database is structured according to a *star schema* design, where data are described in terms of "*facts*", i.e., happenings of interest to be analyzed, and "*dimensions*", i.e. perspectives under which the facts are analyzed. A design based on a star schema enables *multidimensional* analysis (i.e., the analysis of facts seen from different perspectives) and allows the use of many query optimization techniques. PDW includes the following facts (see Figure 1):

- **Process instance executions**
- **Service executions**
- **Behaviors**

Fact data includes such attributes as start and completion times, durations, input process data, and output node data.
These facts can be analyzed based on the following dimensions:

- **Processes and process groups**, to focus on (i.e., restrict the analysis to) facts related to a specific process definition (and possibly a specific version), or to a set of process definitions, or to processes within a group.
- **Services and service groups**, to focus on facts related to a specific service definition, or to a set of service definitions, or to services within a service group.
- **Work nodes**, to focus on facts related to a specific work node definition, or to a set of node definitions.
- **Data Items,** to focus on specific process data items.
- **Resources,** to focus on processes started by, nodes assigned to, or node executed by a specific human or automated resource or group of resources.
- **Time***,* to focus on facts occurred in a certain (fiscal or calendar) time window, or on specific days, weekdays, or hours of the day.
- **Behaviors**, to focus on instances that exhibited a user-defined behavior of interest (details on behaviors are provided below).

PDW users can configure the warehouse in order to limit the amount of data that is loaded. For example, users can limit loaded information to processes belonging to a certain group. In this way, the PDW size remains smaller and queries execute faster.

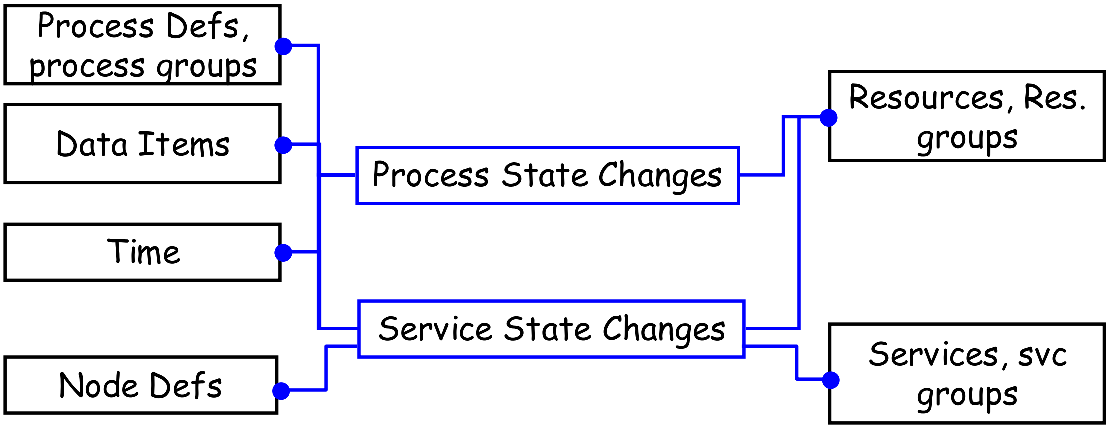


Figure 1 - PDW schema. Facts are depicted with a thicker border, while dimensions have a thin border

Behaviors

One of the most significant and innovative feature of the HPPM intelligent Process Data Warehouse is *behavior analysis*. In fact, a frequent analysis need is that of identifying process instances that exhibit specific behaviors, and to understand the *causes* of such behaviors. Examples of behaviors of interest are supply chain process instances that last more than 20 days, *Expense Approval* process instances that include more than 3 approval cycles, *Claim Management* instances in which node "Examine expenses" was executed by a manager, or processes instances related to order for goods over 20,000$.

The PDW approach is agnostic about the behaviors that users may be interested in analyzing. Indeed, it allows users to define the behaviors to be monitored. PDW will then take care of identifying which processes exhibit a specific behavior and of analyzing them. As shown throughout this document, we use behaviors for a wide variety of purposes. "Reusing" the behavior concepts for several different functionalities simplifies the user interaction with the system since, by getting familiar with the notion of behaviors, users can configure a variety of different analysis, monitoring, and management tasks.

Behaviors are defined by instantiating *behavior templates*. A template is a parametric definition of a behavior, such as "*Instances of process **P** that takes more than **N** days to complete*". In order to define a behavior of interest for a specific process, users simply need to instantiate the template, i.e., provide values for the parameters. From the users' perspective, this is as simple as filling out a form. No coding is needed. Multiple specific behaviors to be monitored (on the same or different processes) can be defined for each behavior type, and a process can be analyzed for multiple behaviors.

PDW includes a large set of predefined behavior templates, to account for the most common monitoring and analysis needs. For example, behaviors can be defined to analyze processes lasting less (more) than a specified duration, including more (less) than *n* activations of a specific work node *WN*, or in which work node WN has (not) been executed by a resource in a group *G*.

Behavior templates are (conceptually) defined by Boolean conditions over process and node execution data available in the warehouse. Templates are implemented by means of SQL statements, that detect behaviors of interest when data are loaded into the warehouse.

Users can add new behavior templates by downloading them from template libraries, made available on the web. This process is automated if the warehouse is used in conjunction with the Business Process Cockpit (that operates analogously to the way antivirus software packages update their virus definition files). If users wish to monitor a kind of behavior that is neither among the predefined ones nor downloadable from web template libraries, they can still specify the behavior template they need, although in this case they would need to define the corresponding condition (i.e., the SQL statement). The occurrence of a behavior is stored as a fact in the warehouse (see Figure 1), so that processes can be also analyzed from the behavior perspective.

By detecting behaviors of interest, analysts can perform multidimensional analysis to understand the causes of "good" and "bad" process executions, and take actions accordingly. In particular, a very useful analysis consists in examining *correlations* among behaviors, i.e., in examining which other behaviors occur when a process instance is affected by a behavior B1. In this way, the effects of B1 on the process can be analyzed. For example, the analyst can define B1 as being processes "started by John Smith" and B2 as processes being "too slow". Behavior analysis can be used to first examine how many processes are "too slow" (say, 15%), and then to examine how many processes among those "started by John Smith" are "too slow" (say, 55%), thereby indicating a cause-effect relationship between John Smith and the process being slow.

PDW analysts can also associate a *value* (or *cost*) to behaviors, to denote the associated benefit or cost. For example, it is possible to say that the fact that when a certain node in a process execution is performed by a unit manager, then a value (cost) of –3 is assigned to the process instance. When the same node is performed by a department manager, then a value (cost) of –2 is assigned, and so on. In this way it is possible to get reports about the combined value (cost) of a set of process executions, by summing the values for each behaviors that a process instance had. Figure 2 shows a chart, obtained by accessing a PDW view with Oracle Discoverer, that shows the total process value (i.e, the sum of the value of the process instances). Data are aggregated based on the week in which the process instances started.

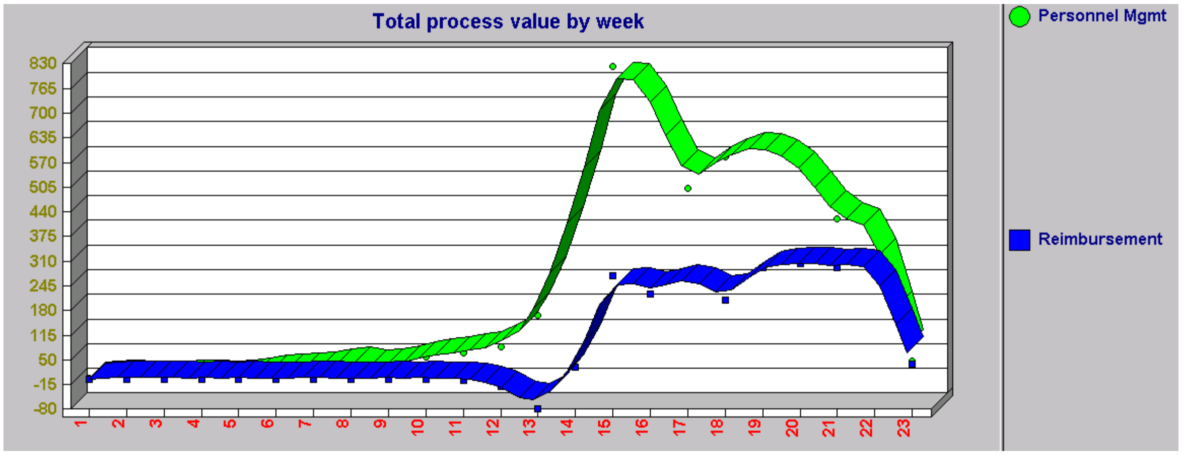


Figure 2 - A chart depicting the total process value, shown by week of the year

Taxonomies

PDW allows the definition of process *taxonomies*. A taxonomy is a user-defined criterion to classify instances of a process depending on their characteristics. Many taxonomies can be defined for the same process, and each taxonomy can have several *categories*. For each taxonomy, a process instance can be in one and only one category at any given time. For example, a taxonomy *outcome* may include the categories *success* and *failure*, while the taxonomy *duration* may include the categories *fast, acceptable, slow,* and *very slow.*

Taxonomies can be defined by specifying the categories that compose the taxonomy. Each category is then associated to a behavior, with the meaning that the process instance is classified in a given taxonomy if the process instance has the corresponding behavior. Taxonomies are flat, that is, there is no hierarchy among categories. Two categories, *Other* and *Error*, are automatically defined by PDW for each taxonomy. Other contains instances that do not fall in any other category within the taxonomy, while Error includes instances belonging to more than one category (PDW does not make any verification about the mutual exclusion of the conditions at process definition time. However, it enforces it at run time.)

For example, Figure 3 shows, for each week, the distribution of process instances within the user-defined "duration" taxonomy, described above.

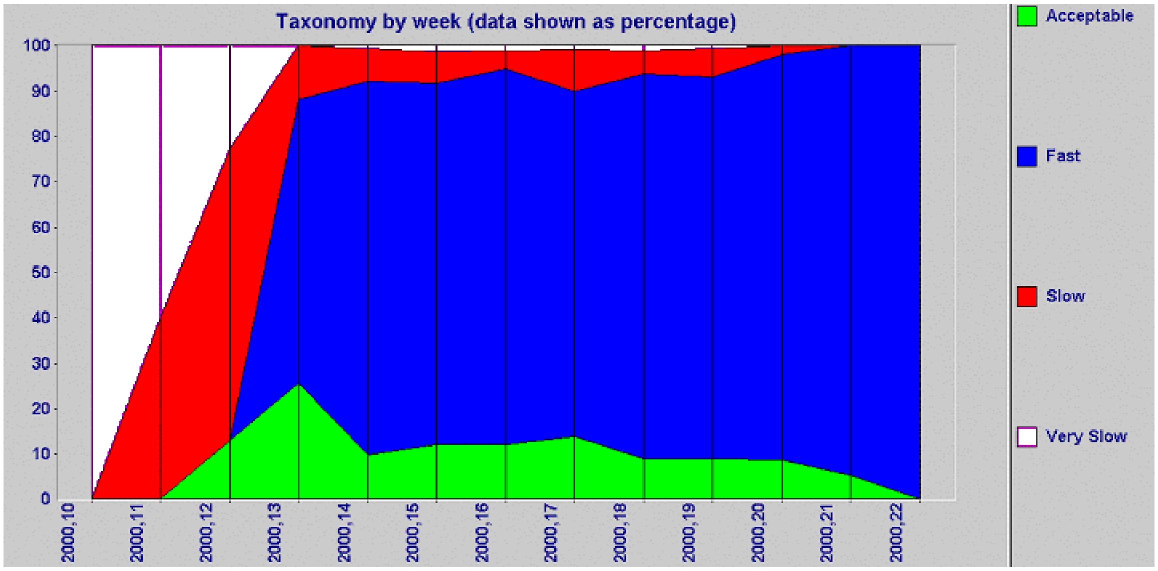


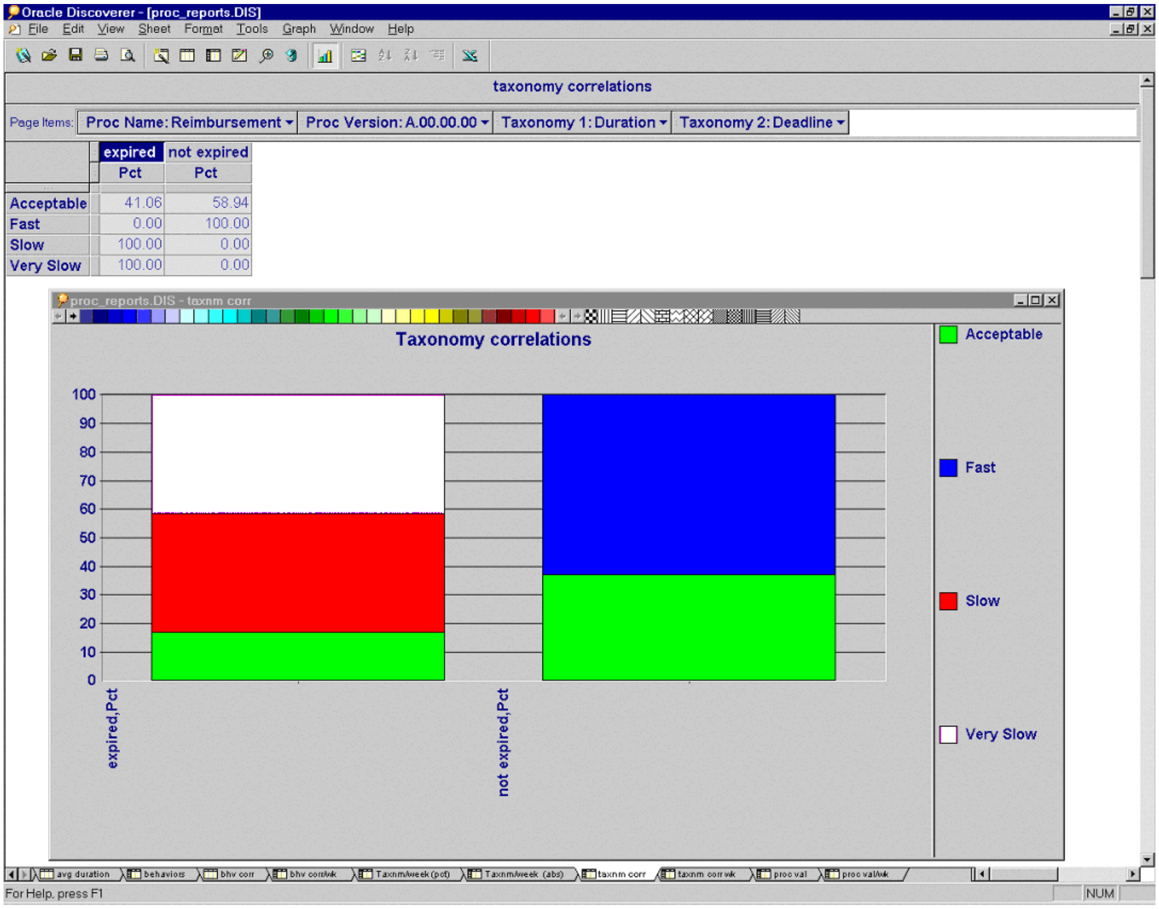
Figure 3 - Distribution of process instances within a taxonomy (shown by week)

Analogously to behaviors, PDW users can also examine correlations among categories of different taxonomies. This kind of analysis is very powerful, since it allows users to easily understand the cause-effect relationships among the categories of the two taxonomies. For example, Figure 4 shows the correlation between the categories of taxonomies *duration* and *deadline* (that describes whether the deadline for node "approve" within process "reimbursement" has expired at least once in process instance execution). Users can examine the process performance distribution, depending on whether the deadline for the "approve" node has expired or not.

Note that, while taxonomy correlations can typically produce reports that are at a higher level of abstraction with respect to behavior correlations (and therefore easier to interpret), they do not replace it. Indeed, users often need to define behaviors that do not belong to specific taxonomies (possibly because they do not generate a partition in the process instance space), and to analyze correlation among these behaviors. Hence, both taxonomy and correlation analysis are useful and needed.

Views and Reports

One of the main PDW design goals was to make it very easy for users to obtain reports from the PDW. For this reason, the PDW includes a large set of views, to account for the large majority of reporting needs (the complete list of views is provided in the appendix). Analysts only need to write simple queries to get the report they need. In addition, PDW provides configuration files for the most common reporting tools, such as Oracle Discoverer or MS Excel, so that using the PDW requires minimum configuration effort. Once PDW and the reporting tool have been installed and data have been loaded, users simply need to import the provided configuration file to view the preconfigured reports, such as the ones shown in the figures above.



taxonomy correlations

	expired	not expired
	Pct	Pct
Acceptable	41.06	58.94
Fast	0.00	100.00
Slow	100.00	0.00
Very Slow	100.00	0.00

Figure 4 - Correlations among categories of different taxonomies

In order to provide acceptable performance while at the same time avoid explosion in disk space usage, most reporting views are not materialized. Instead, a smaller set of materialized views is defined, joining and aggregating data in different ways to support the needs of several reporting views (see Figure 5). Through the "query rewrite" mechanisms provided by the DBMS, PDW automatically uses materialized views (where possible) to speed up queries on the reporting views, reducing the execution time.

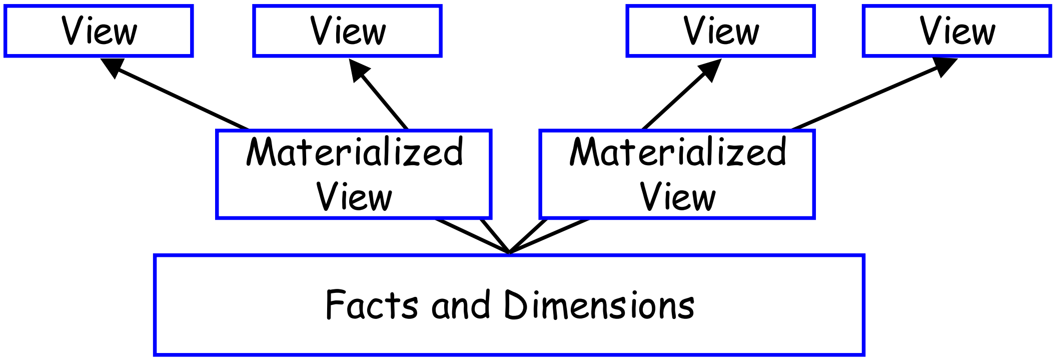


Figure 5 - A small set of materialized views support a large set of reporting views

Process monitoring

Besides providing data and statistics on completed processes, PDW can also provide information on active processes, to enable process monitoring and management, and to provide (high (business) level as well as detailed information about the health not only of the operational system, but also of the overall business.

PDW has a set of monitoring views, that can be queried to get reports on process and system status. For example, PDW includes views that can be used to get the number of active processes, node, and services, and their current duration (i.e., the time elapsed from activation to the time the data has been collected), or the number of active services per node and per resource, or the current execution cost for each process. As an example, Figure 6 shows the number of active service instances for each resource.

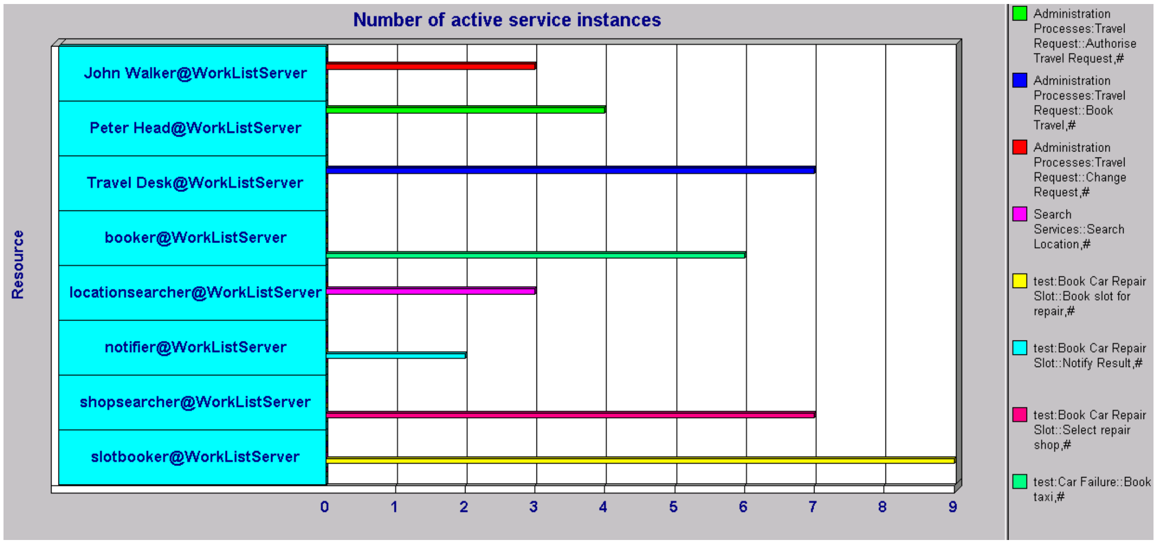


Figure 6 - Report on the number of active services in each resource work queue

Besides reporting several kinds of statistics on active processes, PDW also allows users to define *alerts*. An alert is a condition on active process data that is of particular interest to the analyst, possibly because it corresponds to an undesired situation that needs to be addressed. Like behaviors and taxonomies, alerts allow users to add semantics to information stored in the warehouse, to allow higher-level business process analysis (or monitoring, in this case).

In order to keep the PDW configuration and usage simple, alerts (like taxonomies) are defined by reusing the notion of behaviors: in fact, an alert is characterized by a name and by an associated behavior. When data on active processes are refreshed, PDW detects behaviors (alerts) of interest and stores them into tables that can be queried to get the number and kind of alert. Configuration files for reporting tools already include predefined queries and charts to show alerts.

Value information in behavior definition can be used to indicate the perceived importance of the alert (consistently with the semantics of the value within behavior, lower numbers denote higher importance of the alert since they correspond to costs, i.e., problems).

Populating the HPPM Process Data Warehouse

PDW offers two main options for extracting data from HPPM logs and loading the warehouse: automatic and manual. The options differ in the way the data is *extracted* from the warehouse and is *transferred* to the PDW for loading.

- **Automatic:** periodically, data are automatically extracted from the HPPM logs and transferred to PDW for loading. The interval between subsequent data extraction can be set by the user. No human interaction is needed during the extract, transfer, and load (ETL) process, unless the user wants to manually activate the transfer before the load interval expires. The automatic load mode requires a database link between the PDW and the HPPM audit log database: therefore, it can only be used if both databases are behind the same firewall. In addition, we observe that the level of security is that provided by the DBMS, and no additional mechanisms are provided by PDW to sign, encrypt, or otherwise protect the data during the transfer.
- **Manual:** the user copies the HPPM logs to a file and transfers it (through FTP, disk, or other means) to the PDW for loading. PDW provides applications that extract data from the logs and store them into a DBMS dump file, as well as applications that load data from the dump file into the warehouse.

An important requirement for the ETL procedure is to minimize the impact on the HPPM operations. In fact, in order to extract data, PDW applications need to access the HPPM logs, and therefore to obtain read and write access to the logs, i.e., it needs to obtain exclusive locks on the tables. While PDW applications are accessing HPPM logs, HPPM cannot write the logs, and is forced to wait. This can potentially slow down the HPPM operations. In order to minimize the impact, PDW applications include several features that reduce the time needed to access the HPPM logs. These include:

- **Early commit**: data are copied into HPPM logs *shadow* tables (i.e., tables that have a format similar to that of the HPPM logs tables), and then a commit is performed, to release the locks. This process is typically very fast. During the copy process, data that are no longer needed by HPPM are also purged from the logs (to allow for faster loading next time). Then, the required ETL operations are performed on the shadow tables, without impacting the HPPM operations.
- Database **triggers** are used on tables whose content does not change frequently (such as the process, node, and service definition tables) to automatically copy modifications into temporary tables. Since updates are not frequent, the overhead imposed by triggers is very limited. The benefits are instead considerable, since only the few, newly inserted

tuples need to be copied from the temporary into the shadow tables. Therefore, the copy operations are even faster, reducing the interval in which locks are required.

- **Partition exchange** is the faster way to extract data from the logs, but it can be used only if the HPPM database has been installed with the "partitioning" option. Partition exchange enables swapping between data partitions of different tables. PDW uses partition exchange as follows (see Figure 7): at PDW installation time, it alters HPPM tables to create partitions on those tables. In particular, it creates partitions that contain only data about completed process, node, and service executions. Then, to extract data from the logs, the ETL application simply exchanges the partition with the corresponding (empty) partitions on the shadow tables. This process is extremely fast and does not delay the HPPM operations. Since partition exchange replaces an HPPM partition with an empty one, it cannot be used to extract data that are still needed in the HPPM logs, because HPPM expects to find them there (such as data on active processes). However, the volume of data that can be copied with this technique (including for example data on completed instances) is usually very high, therefore the benefit of partition exchange are considerable.

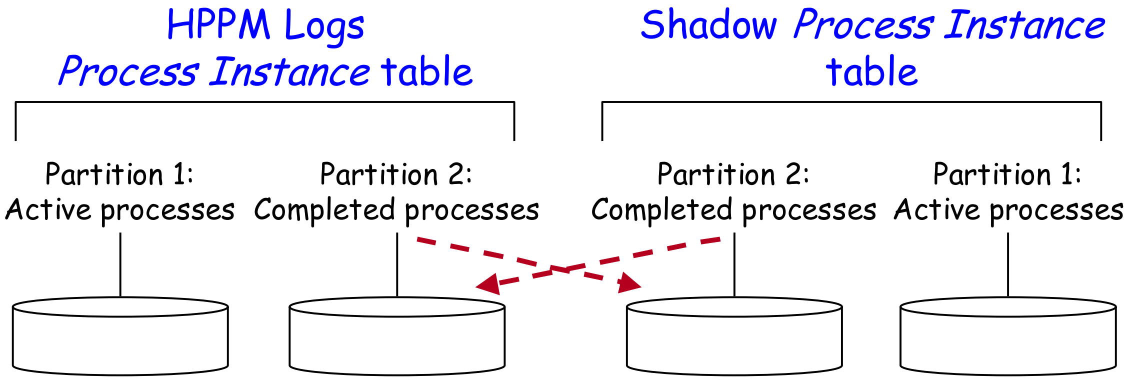

HPPM Logs
Process Instance table
Shadow Process Instance table
Partition 1: Active processes
Partition 2: Completed processes
Partition 2: Completed processes
Partition 1: Active processes

Figure 7 - Partition exchange can be used to extract data on completed executions

Once the data are extracted from the logs and transferred to the PDW (through FTP or database link), they can be cleaned and loaded into the warehouse. The load procedure involves cleaning end ensuring consistency of the data, transforming the data into the PDW format, and detecting behaviors as well as other "semantic" information that can be deduced from the logged data. Finally, the data are inserted into the PDW. The ETL process is summarized in Figure 8.

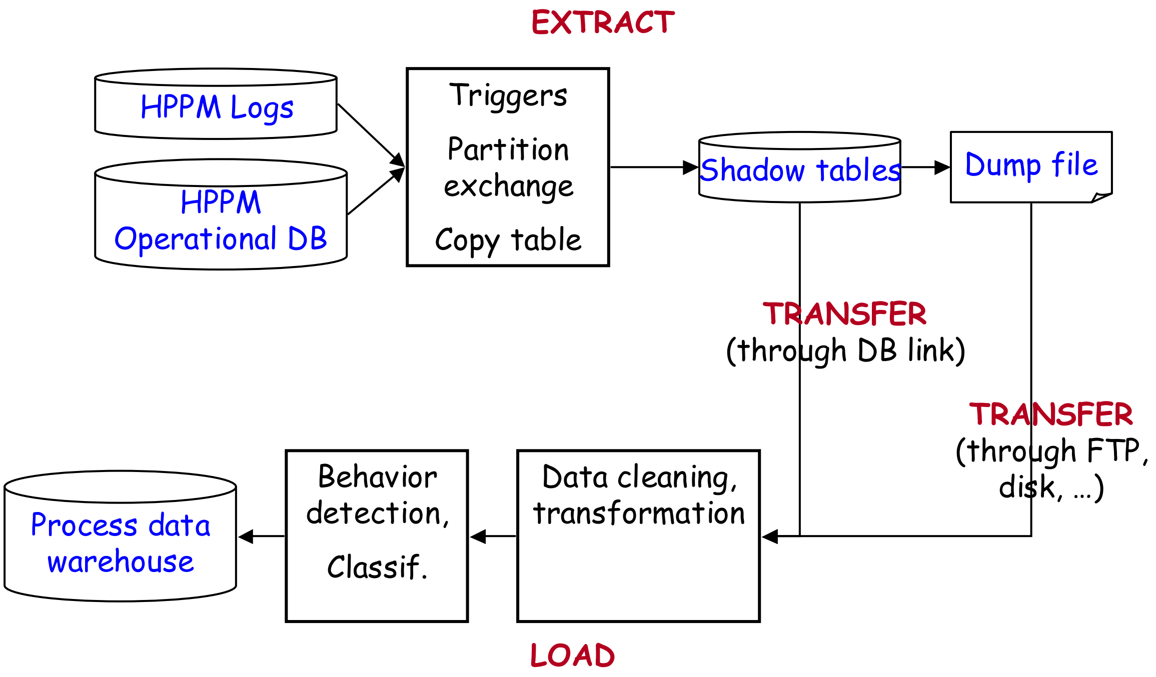


Figure 8 - The ETL process for populating the PDW

APPENDIX A: STRUCTURE OF PDW FACTS AND DIMENSIONS

This appendix details the structure of the tables that contains facts and dimensions.

Process Instance Facts (Table Proc_Inst_F)

This table contains facts on completed process instances.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Process instance identifier (generated by PDW)
PROC_DEF_ID	NUMBER	Unique process definition identifier
START_TIME_ID	DATE	Process start time (rounded up to the hour)
END_TIME_ID	NUMBER	Process end time (rounded up to the hour)
STATE	NUMBER (2)	Final state of the process instance (codes as in the HPPM logs)
INITIATOR_ID	NUMBER	ID of the resource that started the instance
DURATION	VARCHAR2(256)	Process instance duration

		(in days)

Process Data Facts
(Table Proc_Data_Inst_F)

This table lists the input data of each process instance.

COLUMN	TYPE	DESCRIPTION
DATA_DEF_ID	NUMBER	Reference to the data definition identifier.
PROC_INST_ID	NUMBER	Reference to the process instance identifier.
PROC_DEF_ID	NUMBER	Reference to the process definition identifier.
VALUE	VARCHAR2(255)	Value of the data item at process instance start.

Process behavior Facts
(Table Proc_Bhv_Inst_F)

This table lists the behaviors that each instance had.

COLUMN	TYPE	DESCRIPTION
PROC_INST_ID	NUMBER	Process instance identifier (generated by PDW)
PROC_DEF_ID	NUMBER	Process definition identifier
BHV_ID	NUMBER	behavior identifier
START_TIME_ID	DATE	Process start time (rounded up to the hour)

Service Instance Facts
(Table Service_Inst_F)

This table contains facts on service instances executed within processes described in Proc_inst_F.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Service instance identifier (generated by PDW)
PROC_DEF_ID	NUMBER	Process definition identifier
NODE_DEF_ID	NUMBER	Node definition identifier
SERVICE_DEF_ID	NUMBER	Service definition identifier
PROC_INST_ID	NUMBER	Process instance identifier
ACTIVATION_NUMBER	NUMBER	Progressive number of times the service has been activated within this

		node and process instance *(not implemented yet)*
SCHEDULED_TIME_ID	DATE	Time the service instance was scheduled for execution (rounded up to the hour)
ACTIONED_TIME_ID	DATE	Time the service instance was activated by the resource to which it was assigned (rounded up to the hour)
COMPLETED_TIME_ID	DATE	Time the service instance was completed (rounded up to the hour)
STATE	NUMBER (2)	State of the service instance (0=sent, 1=received, 2=actioned, 3=finished, 9=otherwise)
ASSIGNED_TO	NUMBER	ID of the resource to which the service was assigned
EXECUTED_BY	NUMBER	ID of the resource that executed the service (may differ from the one to which it was initially assigned)
SCHEDULED_TO_ACTIONED	NUMBER	Interval elapsed from service scheduling time to service actioning time (in days)
ACTIONED_TO_COMPLETED	NUMBER	Interval elapsed from service actioning time to service completion time (in days)
SCHEDULED_TO_COMPLETED	NUMBER	Interval elapsed from service scheduling time to service completion time (in days)

Node Instance Data (Table Node_Inst_Data_F)

This table contains the values of case packet data items modified by node executions.

COLUMN	**TYPE**	**DESCRIPTION**
DATA_DEF_ID	NUMBER	Reference to the data definition identifier.
NODE_TYPE	NUMBER	Node type. 4= work, 6=route

PROC_DEF_ID	NUMBER	Process definition identifier
NODE_DEF_ID	NUMBER	Node definition identifier
SERVICE_DEF_ID	NUMBER	Service definition identifier (null for data modified by route node)
PROC_INST_ID	NUMBER	Process instance identifier
NODE_INST_ID	NUMBER	Node instance identifier
TIME_ID	DATE	Time the node instance was completed, and the modification applied (rounded up to the hour)
VALUE	VARCHAR2 (255)	Value of the case packet data item after the execution of this node.

Process Dimension
(Table Proc_Defs_D)

This table lists process definitions and the group to which they belong.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Process definition identifier (generated by PDW)
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process Version
PROC_GROUP_NAME	VARCHAR2(512)	Name of the group to which the process belongs

Node Dimension
(Table Node_Defs_D)

This table lists node definitions. It includes all types of nodes. Service information is only filled for work nodes, and it is null otherwise.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Node definition identifier (generated by PDW)
NODE_NAME	VARCHAR2(255)	Node name
PROC_DEF_ID	NUMBER	Identifier of the process definition to which the node belongs
SERVICE_DEF_ID	NUMBER	Identifier of the service definition associated to this node (for work nodes only)
TYPE	NUMBER	Node type 4=work node 6=route node

Service Dimension
(Table Service_Defs_D)

This table lists service definitions and the group to which they belong.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Service definition identifier (generated by PDW)
SERVICE_NAME	VARCHAR2(255)	Service name
SERVICE_VERSION	VARCHAR2(255)	Service definition version
SERVICE_GROUP_NAME	VARCHAR2(255)	Group to which the service belongs
IS_LOCAL	NUMBER (1)	Defines whether the service is local (=1) or global (=0)

Resource Dimension
(Table Resources_D)

This table lists resources and the group to which they belong.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Resource identifier (generated by PDW)
RESOURCE_NAME	VARCHAR2(256)	Resource name
RESOURCE_GROUP_NAME	VARCHAR2(256)	Group to which the resource belongs

Time Dimension
(Table Time_D)

This table lists all different time instants (rounded up to the hour) in which a process or service fact occurred. Time instants are also decomposed by explicitly storing the year, month, day, to which they correspond, as well as other characteristics. Users can also define fiscal dates (see the section on PDW configuration for information on how to specify fiscal dates).

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Identifier of this time instant
YEAR	NUMBER	Year
MONTH	NUMBER	Month
WEEK	NUMBER	Week of the year
DAY	NUMBER	Day of the month
DAYINWEEK	NUMBER	Day of the week (1=Sunday-> 7=Saturday)
HOUR	NUMBER	Hour of the day (0-23)
FISCALYEAR	NUMBER	Fiscal year
FISCALQUARTER	NUMBER	Fiscal quarter
FISCALMONTH	NUMBER	Fiscal month

Behavior Dimension
(Table Proc_Bhv_D)

This table lists the behaviors that have been defined for each process.

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Behavior definition identifier
BHV_NAME	VARCHAR2(255)	behavior name
PROC_DEF_ID	NUMBER	Identifier of the process definition to which the node belongs
VALUE	NUMBER	Value (or cost, if negative) associated to the occurrence of this behavior

Data Dimension
(Table Data_Defs_D)

This table contains the data items defined for each process[1].

COLUMN	TYPE	DESCRIPTION
ID	NUMBER	Data definition identifier (generated by PDW).
DATA_NAME	VARCHAR2 (255)	Name of the data item
DATA_TYPE	VARCHAR2 (255)	0 maps to String 1 maps to StringSeq 2 maps to Integer 3 maps to IntSeq 4 maps to Real 5 maps to RealSeq 6 maps to VarSeq 7 maps to Any
PROC_DEF_ID	NUMBER	Process definition identifier

Arc definitions (Table Arc_Defs)

Arc Definitions
(Table Arc_Defs)

This table lists the arcs within each process flow.

COLUMN	TYPE	DESCRIPTION
PROC_DEF_ID	NUMBER	Identifier of the process definition to which the arc belongs.
SOURCE_NODE_DEF_ID	NUMBER	Reference to the node

[1] Due to the way the HPPM log is structured, data definitions can only be loaded when execution data are also available. Otherwise it is not possible to know which data item has been defined for each process.

		definition identifier that is the source of the arc
DEST_NODE_DEF_ID	NUMBER	Reference to the node definition identifier that is the destination of the arc

APPENDIX B: PDW REPORTING VIEWS

PDW includes a wide set of views, that can satisfy many reporting needs. They include:

- *Process-oriented views*, providing reports about process executions.
- *Node-oriented views*, providing reports about node and service executions.
- *Behavior analysis views*, that provide statistics about behaviors of interest and help identifying the causes of such behaviors.
- *Resource analysis views*, to analyze the performance and efficiency of resources, both in absolute terms and relative to other resources.

Process-oriented views

Process Statistics (View Proc_Stats_V)

Statistical data about process instance executions. For each process definition and version, the view reports the number of completed instance executions, as well as the average duration and the standard deviation of the duration for such instances.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations	NUMBER	Total number of instances executed
AVG_Duration	NUMBER	Average duration of the completed instances
STDDEV_Duration	NUMBER	Standard deviation of the duration of completed instances

Process Statistics by Initiator (View Proc_Stats_Initiator_V)

Statistical data about process instance executions, depending on the resource that started the process (the initiator). For each process definition version, and initiator, the view reports the number of completed instance executions as well as the average duration and the standard deviation of the duration of such executions.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations	NUMBER	Total number of completed instances started by this initiator
AVG_Duration	NUMBER	Average duration of the completed instances started by this initiator
STDDEV_Duration	NUMBER	Standard deviation of the duration of completed instances started by this initiator
INITIATOR	VARCHAR2(256)	Name of the resource who started the process instances

In the following we provide two sample reports that can be obtained by querying this view. The first displays, for each process, statistical data depending on the initiator. The second report shows, for each initiator, statistical data about the processes started by that initiator.

Process Statistics by starting day (View Proc_Stats_C_Date_V)
Statistical data about process instance executions (as in the proc_stats_V view) depending on the calendar date (i.e., year, month, and day of the month the process instances was started).

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations	NUMBER	Total number of instances executed
AVG_Duration	NUMBER	Average duration of the completed instances
STDDEV_Duration	NUMBER	Standard deviation of the duration of completed instances
C_Year	NUMBER (4)	Calendar year to which the data refers
C_Month	NUMBER (2)	Calendar month to which the data refers
C_Day	NUMBER (2)	Calendar day of the month to which the data refers

Rollup_Year	NUMBER	Rollup attribute for C_year
Rollup_Month	NUMBER	Rollup attribute for C_month
Rollup_Day	NUMBER	Rollup attribute for C_day

This view includes both summary and detail information. In fact, it can show data for each process definition and each day. In addition, it can aggregate data for all days of the month (to get monthly reports), for all months (to get yearly reports), and for all years (to get complete historical reports).
Aggregations are controlled by the "Rollup_XX" attributes. Tuples with all the rollup attributes equal to 0 contain data about a specific day of the year. Tuples with a rollup attribute "Rollup_XX=1" returns aggregate data for the specified attribute. For example, querying the view with the condition Rollup_Day=1 (with Rollup_Month=0 and Rollup_year=0) returns monthly reports. Tuples with Rollup_Day=1, Rollup_Month=1, but Rollup_year=0 provide yearly reports, while tuples with all rollup attributes equal to 1 provide overall statistics (just like the *proc_stats_v* view).
Note that aggregation can only be done from the *day* to the *year:* while it is possible to aggregate days (to have data at the year and month level), it is not possible to have data that is month-independent but day-specific. In other words, it is not possible to get information abut, for example, process statistics on the 26th of the month without specifying a specific year and month. On the contrary, it is possible to have statistics for June 2001, without specifying the day. Explained in yet another (more practical) way, if rollup_day=0, then also rollup_month and rollup_year are equal to 0.

Process Statistics by starting fiscal period (View Proc_Stats_F_Date_V)

Statistical data about process instance executions (as in the proc_stats_V view) depending on the fiscal period (i.e., fiscal year, fiscal quarter, and fiscal month in which the process instances was started).

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations	NUMBER	Total number of instances executed
AVG_Duration	NUMBER	Average duration of the completed instances
STDDEV_Duration	NUMBER	Standard deviation of the duration of completed instances
F_Year	NUMBER (4)	Fiscal year to which the data refers
F_Quarter	NUMBER (2)	Fiscal quarterto which the

		data refers
F_month	NUMBER (2)	Fiscal month to which the data refers
Rollup_F_Year	NUMBER	Rollup attribute for F_year
Rollup_F_Quarter	NUMBER	Rollup attribute for F_quarter
Rollup_F_Month	NUMBER	Rollup attribute for F_month

Like the previous view, this one also can show both detailed and aggregate data. Note that aggregation can only be done from the *month* to the *year* (see description of the above views for details about the order of aggregation and its meaning).

Process Statistics by Weekday and Hour (View Proc_Stats_WeekdayHr_V)

Analogous to the reports above, but providing a report focused on the day of the week and hour of the day.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations	NUMBER	Total number of instances executed
AVG_Duration	NUMBER	Average duration of the completed instances
STDDEV_Duration	NUMBER	Standard deviation of the duration of completed instances
Dayinweek	NUMBER (1)	Day of the week to which the data refers (0=Sunday)
Hour	NUMBER (1)	Hour of the day to which the data refers (0-23)

Process Weekly Trend (View Proc_Stats_Week_V)

Statistical data about process instance executions depending on year and week in the year the process instances started.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations	NUMBER	Total number of instances executed
AVG_Duration	NUMBER	Average duration of the

		completed instances
STDDEV_Duration	NUMBER	Standard deviation of the duration of completed instances
Year	NUMBER (4)	Year to which the data refers
Week	NUMBER (2)	Week of the year to which the data refers

Behaviors

Statistics on process behaviors (View Proc_Bhv_V)

This view summarizes, for each process, the number of instances that exhibited a specific behavior (e.g., *duration>10 days*, or *Managers involved in instance execution*), along with the hit ratio, i.e., the percentage of instances characterized by the behavior. For example, it can be queried to discover how many instances (and what percentage of instances) of a process had the "*Managers involved in instance execution*" behavior.

COLUMN	**TYPE**	**DESCRIPTION**
PROC_DEF_ID	NUMBER	Unique process identifier
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process version
BHV_DEF_ID	NUMBER	Identifier of the behavior definition
BHV_NAME	VARCHAR2(256)	Name of the Behavior definition
NUM_INSTANCES	NUMBER	Number of instances that had the behavior.
PERCENTAGE	NUMBER	Percentage of instances that had this behavior

Behavior Statistics by week (View Proc_Bhv_Week_V)

This view reports on the number of process instances that had a certain behavior, depending on the process instance starting year and week of the year.

COLUMN	**TYPE**	**DESCRIPTION**
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Bhv_Def_ID	NUMBER	Identifier of the behavior definition
Bhv_Name	VARCHAR2(256)	Name of the Behavior

		definition
Num_Instances	NUMBER	Number of instances that had the behavior.
Percentage	NUMBER	Percentage of instances that had this behavior
C_Year	NUMBER(4)	Calendar year to which the data is related
C_Week	NUMBER(2)	Calendar week of the year to which the data is related

Correlation among Behaviors (View Proc_Bhv_Corr_V)

This is a very powerful view that allows discovering correlation among behaviors. Each tuple contains information about the occurrence of a behavior (bhv2) in instances of a process, and then also give information about the correlated behaviors, i.e., how many instances have both bhv1 and bhv2. For example, a tuple can give information about the number and percentage of TER process instances that had behavior *Managers involved in instance execution* (just like Proc_Bhv_V), but also about the number and percentage of instances that have both the *Managers involved in instance execution* and the *duration>10 days* behaviors. In this way the analyst can analyze behavior bhv2, by seeing how often it occurs in general and how often it occurs when bhv1 also occurs, to discover possible correlations and therefore gain additional information.

COLUMN	**TYPE**	**DESCRIPTION**
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Bhv1_Def_ID	NUMBER	Identifier of the behavior 1 definition
Bhv1_Name	VARCHAR2(256)	Name of the Behavior 1 definition
Bhv2_Def_ID	NUMBER	Identifier of the behavior 2 definition
Bhv2_Name	VARCHAR2(256)	Name of the Behavior 2 definition
Bhv2_Num_Instances	NUMBER	Number of instances that had behavior 2.
Bhv2_Percentage	NUMBER	Percentage of instances that had behavior 2
Corr_Num_Instances	NUMBER	Number of instances that had both behavior 2 and bhv 1.
Corr_Percentage	NUMBER	Percentage of instances that had behavior 2 among those who also had bhv 1 (equal to

		Corr_Num_Instances/ instances that had bhv 1)

Correlation among Behaviors by week (View Proc_Bhv_Corr_Week_V)

This view is analogous to the Proc_Bhv_Corr_V, but shows detailed information depending on the week in which instances process started, as opposed to being limited to showing aggregate results.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Bhv1_Def_ID	NUMBER	Identifier of the behavior 1 definition
Bhv1_Name	VARCHAR2(256)	Name of the Behavior 1 definition
Bhv2_Def_ID	NUMBER	Identifier of the behavior 2 definition
Bhv2_Name	VARCHAR2(256)	Name of the Behavior 2 definition
Bhv2_Num_Instances	NUMBER	Number of instances that had behavior 2.
Bhv2_Percentage	NUMBER	Percentage of instances that had behavior 2
Corr_Num_Instances	NUMBER	Number of instances that had both behavior 2 and bhv 1.
Corr_Percentage	NUMBER	Percentage of instances that had behavior 2 among those who also had bhv 1 (equal to Corr_Num_Instances/ instances that had bhv 1)
C_Year	NUMBER(4)	Calendar year to which the data is related
C_Week	NUMBER(2)	Calendar week of the year to which the data is related

Statistics on process taxonomies (View Proc_Taxonomy_V)

This view contains taxonomy data: for each process definition and each taxonomy, it shows how many instances belong to each taxonomy, and which percentage of instances is in each taxonomy.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Taxonomy	VARCHAR2(256)	Process taxonomy
Category	VARCHAR2(256)	Category to which the data refers
Num_Instances	NUMBER	Number of instances in this category
Percentage	NUMBER	Percentage of instances in this category

Taxonomy Statistics by week (View Proc_Taxonomy_Week_V)

This view contains data about the weekly trend in process taxonomies. For each process and taxonomy, the view describes the number and percentage of instances in each category. Data can be aggregated at the year level by setting the rollup attribute for week (rollup_c_week) to 1, and can be aggregated further by setting rollup_c_year to 1, thereby getting the same results provided by the proc_taxonomy_v view.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Taxonomy	VARCHAR2(256)	Process taxonomy
Category	VARCHAR2(256)	Category to which the data refers
Num_Instances	NUMBER	Number of instances in this category
Percentage	NUMBER	Percentage of instances in this category
C_Year	NUMBER(4)	Calendar year to which the data is related
C_Week	NUMBER(2)	Calendar week of the year to which the data is related
Rollup_C_Year	NUMBER	Rollup attribute for year
Rollup_C_Week	NUMBER	Rollup attribute for week

Correlation among Taxonomies (View Proc_Taxonomies_Corr_V)

This is a very powerful view that allows discovering correlation among two different taxonomies. Each tuple contains information about the number of instances within a category category_2 of taxonomy_2, about the percentage of the instances in this category, about the number of instances that are classified both in (taxonomy_1,category_1) and in (taxonomy_2,category_2), and finally about the percentage of instances classified in (taxonomy_2,category_2) among those classified in (taxonomy_1,category_1).

COLUMN	**TYPE**	**DESCRIPTION**
PROC_DEF_ID	NUMBER	Unique process identifier
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process version
TAXONOMY_1	VARCHAR2(256)	Process taxonomy
CATEGORY_1	VARCHAR2(256)	Category to which the data refers
TAXONOMY_2	VARCHAR2(256)	Process taxonomy
CATEGORY_2	VARCHAR2(256)	Category to which the data refers
NUM_INSTANCES_2	NUMBER	Number of instances in category 2.
CORR_NUM_INSTANCES	NUMBER	Number of instances that are both in (taxonomy1, category1) and in (taxonomy2, category2).
PERCENTAGE_2_ON_TOT	NUMBER	Percentage of instances in in (taxonomy2, category2)
PERCENTAGE_2_ON_1	NUMBER	Percentage of instances in in (taxonomy2, category2) among the instances that are in (taxonomy1, category1)

Correlation among Taxonomies (View Proc_Taxonomies_Corr_Week_V)

This view is analogous to the Proc_Taxonomies_Corr_V, but shows detailed information depending on the week in which instances process started, as opposed to being limited to showing aggregate results.

COLUMN	**TYPE**	**DESCRIPTION**
PROC_DEF_ID	NUMBER	Unique process identifier
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process version
TAXONOMY_1	VARCHAR2(256)	Process taxonomy
CATEGORY_1	VARCHAR2(256)	Category to which the data refers
TAXONOMY_2	VARCHAR2(256)	Process taxonomy
CATEGORY_2	VARCHAR2(256)	Category to which the data refers
NUM_INSTANCES_2	NUMBER	Number of instances in category 2.

CORR_NUM_INSTANCES	NUMBER	Number of instances that are both in (taxonomy1, category1) and in (taxonomy2, category2).
PERCENTAGE_2_ON_TOT	NUMBER	Percentage of instances in in (taxonomy2, category2)
PERCENTAGE_2_ON_1	NUMBER	Percentage of instances in in (taxonomy2, category2) among the instances that are in (taxonomy1, category1)
C_Year	NUMBER(4)	Calendar year to which the data is related
C_Week	NUMBER(2)	Calendar week of the year to which the data is related

Value statistics (View Proc_Value_V)

This view reports statistical information on the value of process executions, as determined by looking at the behaviors.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_instances	NUMBER	Number of instances considered in the computations (equal to the number of instances for which at least one behavior has been detected)
Total_value	number	Total value (sum of the value the process instances)
avg_value	number	average value (average of the value of the process instances)
max_value	number	maximum value among the process instances
min_value	number	Minimum value among the process instances
stddev_value	number	Standard deviation of the value of the process instances

Value statistics by week

(View Proc_Value_Week_V)

This view reports statistical information on the value of process executions, as determined by looking at the behaviors, depending on the process instance starting year and week of the year.

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_instances	NUMBER	Number of instances considered in the computations (equal to the number of instances for which at least one behavior has been detected)
Total_value	number	Total value (sum of the value the process instances started in the specified year and week)
avg_value	number	average value (average of the value of the process instances started in the specified year and week)
max_value	number	maximum value among the process instances started in the specified year and week
min_value	number	minimum value among the process instances started in the specified year and week
stddev_value	number	Standard deviation of the value of the process instances started in the specified year and week
C_Year	NUMBER(4)	Calendar year to which the data is related
C_Week	NUMBER(2)	Calendar week of the year to which the data is related

Node-oriented views

Service Statistics (View Service_Stats_V)

Statistical data about service instance executions. For each service definition, the view reports the number of completed instance executions, as well as the average time and the standard deviation of the time elapsed from: scheduling to actioning, scheduling to completion, and actioning to completion.

COLUMN	TYPE	DESCRIPTION
Service_Def_ID	NUMBER	Unique service definition identifier
Service_name	VARCHAR2(512)	Service name
Service_group_name	VARCHAR2(512)	Service group name
Num_Activations	NUMBER	Total number of instances executed
AVG_scheduled_to_actioned	NUMBER	Average time elapsed from service scheduling time to service actioning time
STDDEV_scheduled_to_actioned	NUMBER	Standard deviation of time elapsed from service scheduling time to service actioning time
AVG_scheduled_to_completed	NUMBER	Average time elapsed from service scheduling time to service completion time
STDDEV_scheduled_to_completed	NUMBER	Standard deviation of time elapsed from service scheduling time to service completion time
AVG_actioned_to_completed	NUMBER	Average time elapsed from service actioning time to service completion time
STDDEV_actioned_to_completed	NUMBER	Standard deviation of time elapsed from service actioning time to service completion time

Node Statistics (View Node_Stats_V)

Statistical data about work node instance executions (or, seen from another perspective, statistics on service executions grouped by node. In fact, every time a work node is executed, the associated service is executed). The view reports on the number (total, avg, and stddev) of completed service instance executions, as well as the average time and the standard deviation of the time elapsed from: scheduling to actioning, scheduling to completion, and actioning to completion. Tuples with rollup_node=1 contain information aggregated at the process level, without node-level distinctions. Finally, tuples with rollup_proc=1 contain process and node-independent information

COLUMN	TYPE	DESCRIPTION
Proc_Def_ID	NUMBER	Unique process identifier

Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Node_Def_ID	NUMBER	Unique node definition identifier
Node_name	VARCHAR2(255)	Node name
Service_Def_ID	NUMBER	Unique service definition identifier
Service_name	VARCHAR2(255)	Service name
Service_group_name	VARCHAR2(255)	Service group name
AVG_scheduled_to_actioned	NUMBER	Average time elapsed from service scheduling time to service actioning time
STDDEV_scheduled_to_actioned	NUMBER	Standard deviation of time elapsed from service scheduling time to service actioning time
AVG_scheduled_to_completed	NUMBER	Average time elapsed from service scheduling time to service completion time
STDDEV_scheduled_to_completed	NUMBER	Standard deviation of time elapsed from service scheduling time to service completion time
AVG_actioned_to_completed	NUMBER	Average time elapsed from service actioning time to service completion time
STDDEV_actioned_to_completed	NUMBER	Standard deviation of time elapsed from service actioning time to service completion time
Total_num_activations	NUMBER	Total number of activations for this service within the specified node and process (i.e., sum through all the instances)
Avg_num_activations	NUMBER	Average number of activations for this service within the specified node and process
Stddev_num_activations	NUMBER	Stddev of the number of activations for this service within the

		specified node and process
Rollup_proc	NUMBER	Rollup attribute for the process
Rollup_node	NUMBER	Rollup attribute for the node

Resource-Oriented Views

Basic resource/process statistics (Res_stats_Proc_V view)

Statistical data about process executions started by a given initiator. For each initiator, the view shows how many instances of a process have been started, the percentage of instances started by this resource, statistical parameters about instances started by this resource, and statistical parameters about instances of this process (in general, i.e., not only those started by the resource – this allows easy comparison).

COLUMN	TYPE	DESCRIPTION
Resource_ID	NUMBER	Unique resource identifier
RESOURCE_NAME	VARCHAR2(256)	Name of the resource who started the process instances
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Activations_BY_RES	NUMBER	Total number of completed instances started by this initiator
Percentage	NUMBER	Percentage of instances started by this resource
Num_Activations	NUMBER	Total number of instances of this process
AVG_Duration_by_res	NUMBER	Average duration of the instances started by this initiator
STDDEV_Duration_by_res	NUMBER	Standard deviation of the duration of instances started by this initiator
AVG_Duration	NUMBER	Average duration of instances of this process
STDDEV_Duration	NUMBER	Standard deviation of the duration of of instances of this process

Basic resource/service statistics (Res_stats_Service_A_V view)

Statistical data about services assigned to a certain resource. For each resource, the view shows how many instances of a service have been started, the percentage of instances started by this resource, statistical parameters about instances started by this resource, and statistical parameters about instances of this service (in general, i.e., not only those started by the resource – this allows easy comparison).

COLUMN	TYPE	DESCRIPTION
Resource_ID	NUMBER	Unique resource identifier
RESOURCE_NAME	VARCHAR2(256)	Name of the resource to whom the instance was assigned
Service_Def_ID	NUMBER	Unique service definition identifier
Service_name	VARCHAR2(512)	Service name
Service_group_name	VARCHAR2(512)	Service group name
Num_Activations_by_res	NUMBER	Total number of service instances assigned to this resource
Percentage	NUMBER	Percentage of service instances assigned to this resource
Num_Activations	NUMBER	Total number of service instances
AVG_sched_to_act_by_res	NUMBER	Average time elapsed from service scheduling time to service actioning time (for instances assigned to this resource)
STDDEV_sched_to_act_by_res	NUMBER	Standard deviation of time elapsed from service scheduling time to service actioning time (for instances assigned to this resource)
AVG_sched_to_comp_by_res	NUMBER	Average time elapsed from service scheduling

		time to service completion time (for instances assigned to this resource)
STDDEV_sched_to_comp_by_res	NUMBER	Standard deviation of time elapsed from service scheduling time to service completion time (for instances assigned to this resource)
AVG_act_to_comp_by_res	NUMBER	Average time elapsed from service actioning time to service completion time (for instances assigned to this resource)
STDDEV_act_to_comp_by_res	NUMBER	Standard deviation of time elapsed from service actioning time to service completion time (for instances assigned to this resource)
AVG_sched_to_act	NUMBER	Average time elapsed from service scheduling time to service actioning time
STDDEV_sched_to_act	NUMBER	Standard deviation of time elapsed from service scheduling time to service actioning time
AVG_sched_to_comp	NUMBER	Average time elapsed from service scheduling time to service completion time
STDDEV_sched_to_comp	NUMBER	Standard deviation of time elapsed from service scheduling time to service completion time
AVG_act_to_comp	NUMBER	Average time

		elapsed from service actioning time to service completion time
STDDEV_act_to_comp	NUMBER	Standard deviation of time elapsed from service actioning time to service completion time

Basic resource/service statistics (Res_stats_Service_A_V view)

Just like the above view, but this focuses on services *executed by* (as opposed to *assigned to*) a resource.

PDW MONITORING VIEWS

PDW includes a wide set of views that can be used for monitoring active processes and nodes. They include:

- *Process-oriented views*, providing reports about process executions.
- *Node-oriented views*, providing reports about node and service executions.
- *Behavior analysis views*, that provide statistics about behaviors of interest and help identifying the causes of such behaviors.
- *Resource analysis views*, to analyze the performance and efficiency of resources, both in absolute terms and relative to other resources.

Statistics on Active Processes (View Active_Proc_Stats_V)

Statistical data about active process instances. For each process definition and version, the view reports the number of active instances, as well as the average and maximum time elapsed from activation for such instances.

COLUMN	**TYPE**	**DESCRIPTION**
Proc_Def_ID	NUMBER	Unique process identifier
Proc_Name	VARCHAR2(512)	Process name
Proc_Version	VARCHAR2(512)	Process version
Num_Active_procs	NUMBER	Total number of active instances
AVG_Duration	NUMBER	Average time elapsed since the instance activation

Max_Duration	NUMBER	Maximum time elapsed since the instance activation

Statistics on Active Services (View Active_Service_Stats_V)

Statistical data about active service instances. For each service definition, the view reports the number of active instances, as well as the average and maximum time elapsed from scheduling, from actioning, and from scheduling to actioning (the second and third are non null only if the service has been already actioned).

COLUMN	TYPE	DESCRIPTION
SERVICE_DEF_ID	NUMBER	Unique service definition identifier
SERVICE_NAME	VARCHAR2(512)	Service name
SERVICE_GROUP_NAME	VARCHAR2(512)	Service group name
NUM_ACTIVE	NUMBER	Total number of active service instances
AVG_SCHEDULED_TO_ACTIONED	NUMBER	Average time elapsed from service scheduling time to service actioning time
MAX_SCHEDULED_TO_ACTIONED	NUMBER	maximum time elapsed from service scheduling time to service actioning time
AVG_SCHEDULED_TO_CHECKPOINT	NUMBER	Average time elapsed from service scheduling time to the instant the log data was collected (called checkpoint time)
MAX_SCHEDULED_TO_CHECKPOINT	NUMBER	maximum time elapsed from service scheduling time to checkpoint time
AVG_ACTIONED_TO_CHECKPOINT	NUMBER	Average time elapsed from service actioning time to checkpoint time
MAX_ACTIONED_TO_CHECKPOINT	NUMBER	maximum of time elapsed from service actioning time to checkpoint time

Statistics on Active Nodes (View Active_Node_Stats_V)

Statistical data about active work node instances (or, seen from another perspective, statistics on service instances grouped by node. In fact, every time a work node is executed, the associated service is executed). The view reports on the number of active instances of the node, as well as the average and maximum time elapsed from: scheduling to actioning, scheduling to checkpoint, and actioning to checkpoint.

COLUMN	**TYPE**	**DESCRIPTION**
PROC_DEF_ID	NUMBER	Unique process identifier
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process version
NODE_DEF_ID	NUMBER	Unique node definition identifier
NODE_NAME	VARCHAR2(255)	Node name
SERVICE_DEF_ID	NUMBER	Unique service definition identifier
SERVICE_NAME	VARCHAR2(255)	Service name
SERVICE_GROUP_NAME	VARCHAR2(255)	Service group name
NUM_ACTIVE	NUMBER	Number of active instances of this node
AVG_SCHEDULED_TO_ACTIONED	NUMBER	Average time elapsed from service scheduling time to service actioning time
MAX_SCHEDULED_TO_ACTIONED	NUMBER	Standard deviation of time elapsed from service scheduling time to service actioning time
AVG_SCHEDULED_TO_CHECKPOINT	NUMBER	Average time elapsed from scheduling time to the instant the log data was collected (called checkpoint time)
MAX_SCHEDULED_TO_CHECKPOINT	NUMBER	maximum time elapsed from scheduling time to checkpoint time
AVG_ACTIONED_TO_CHECKPOINT	NUMBER	Average time elapsed from actioning time to checkpoint time
MAX_ACTIONED_TO_CHECKPOINT	NUMBER	maximum of time elapsed from actioning time to

		checkpoint time

Basic resource/active service statistics (Active_Res_stats_Service_A_V view)

Statistical data about active services assigned to a certain resource. For each resource, the view shows how many instances of a service are active as well as statistical data about execution time, analogously to what shown in the active_service_stats_v view, but grouped by resource.

COLUMN	TYPE	DESCRIPTION
RESOURCE_ID	NUMBER	Unique resource identifier
RESOURCE_NAME	VARCHAR2(256)	Name of the resource to whom the instance was assigned
SERVICE_DEF_ID	NUMBER	Unique service definition identifier
SERVICE_NAME	VARCHAR2(512)	Service name
SERVICE_GROUP_NAME	VARCHAR2(512)	Service group name
NUM_ACTIVE	NUMBER	Total number of service instances assigned to the resource
AVG_SCHEDULED_TO_ACTIONED	NUMBER	Average time elapsed from service scheduling time to service actioning time (for instances assigned to this resource)
MAX_SCHEDULED_TO_ACTIONED	NUMBER	Standard deviation of time elapsed from service scheduling time to service actioning time (for instances assigned to this resource)
AVG_SCHEDULED_TO_CHECKPOINT	NUMBER	Average time elapsed from service scheduling time to the instant the log data was collected (for instances assigned to this resource).

MAX_SCHEDULED_TO_CHECKPOINT	NUMBER	maximum time elapsed from service scheduling time to checkpoint time (for instances assigned to this resource).
AVG_ACTIONED_TO_CHECKPOINT	NUMBER	Average time elapsed from service actioning time to checkpoint time (for instances assigned to this resource).
MAX_ACTIONED_TO_CHECKPOINT	NUMBER	maximum of time elapsed from service actioning time to checkpoint time(for instances assigned to this resource).

Alerts - Aggregate (view Alerts_aggregate_v)

This view gives information about the number of instances for which an alert is active. The information is grouped by process definition and by alert.

COLUMN	TYPE	DESCRIPTION
PROC_DEF_ID	NUMBER	Unique process identifier
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process version
ALERT_ID	NUMBER	Unique alert identifier
ALERT_NAME	VARCHAR2(256)	name of the alert
NUM_INSTANCES	NUMBER	number of active instances for which the alert is on

Alerts - details (view Alerts_Details_v)

This view lists the instances that have active alerts.

COLUMN	TYPE	DESCRIPTION
PROC_DEF_ID	NUMBER	Unique process identifier
PROC_NAME	VARCHAR2(512)	Process name
PROC_VERSION	VARCHAR2(512)	Process version
PROC_INST_ID	NUMBER	identifier of the instance for which the alert is active
ALERT_ID	NUMBER	Unique alert identifier

ALERT_NAME	VARCHAR2(256)	name of the alert
NUM_INSTANCES	NUMBER	number of active instances for which the alert is on