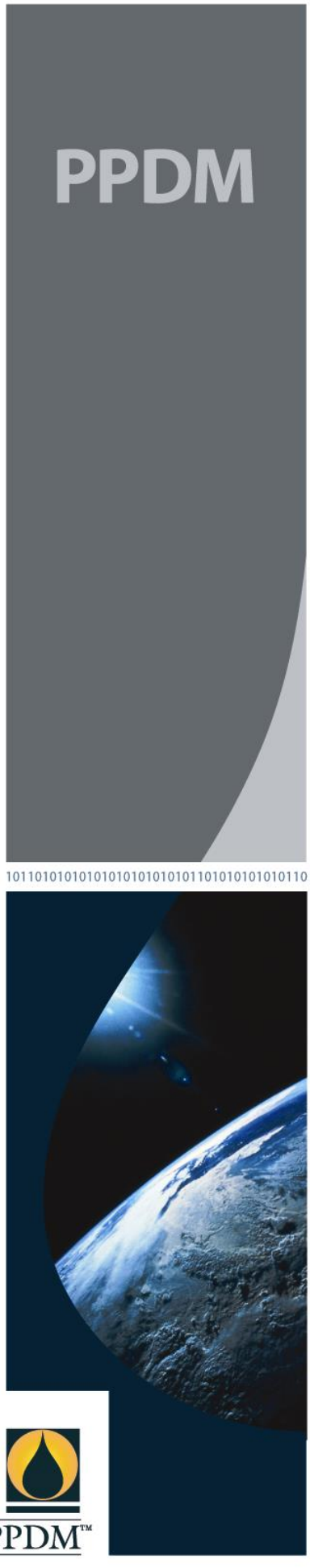
PPDM Association

Reserves

Reference Guide

Last updated for PPDM 3.7



Developed for the PPDM Association by
Robert Gair (Independent)
Trudy Curtis (PPDM CIO)
Wes Baird (dataMatters)

Table Of Contents

About This Document

This reference guide has been prepared to help managers, analysts, database administrators, programmers, data managers, and users understand how to use the Reserves Data Module in PPDM 3.7. Readers at many levels, from managerial to technical implementers will benefit from reading various sections of this document. General, high-level business information is contained at the beginning of the document, with each section becoming progressively more technical and detailed.

Sometimes the terms we use in this and other PPDM documents need to be defined. We provide definitions in a separate Glossary, which you can obtain from the PPDM Association.

This reference guide contains the following sections:

- Introduction

  Provides an executive overview of the PPDM Model as it pertains to Reserves.

- Business Process Overview

  Summarizes Reserves and provides examples of related business processes.

- Integration

  Discusses how Reserves is integrated with the other PPDM Business Modules and provides information about related references guides.

- Model Overview

  Includes the entity relationship diagram and discusses the use of Reserves Module tables in the Data Model.

- Tables and Columns – Reserves

  Identifies the data model tables for the Reserves Module, how they should be used, what they contain, and recommends how they should be used. This section should be used in conjunction with the PPDM Table Report available for download from the PPDM Web Site (www.ppdm.org).

- Implementation Considerations

  Discusses issues related to implementing the PPDM model, architectural methodologies used in design, or special considerations for implementation that are not related to a specific table.

- Frequently Asked Questions

  Addresses technical and business questions about the Reserves Module.

- Appendix A – Sample Queries

Provides example queries with the appropriate SQL scripts that illustrate uses of the model based on the Business Requirements Document.

Introduction

Assets are the foundation on which E&P companies are built. The business of an E&P company is to acquire assets, increase their value through exploration and development activities, earn revenue through production, and dispose of assets once there is no further opportunity to add value to, or receive value from them. In today's increasingly competitive environment it's critical that organizations recognize and respond to both assets and business areas that are not contributing to the organization's profitable growth. Assets that are no longer profitable or that fall outside the corporate areas of expertise are candidates for disposition. Business areas that are not competitive represent opportunities for refocusing resources.

Common strategies among the top oil and gas firms include aggressively managing their asset portfolio, restructuring their asset base, focusing on core strengths and expanding high return core businesses and improving the performance of under-performing business areas by reducing costs. Business drivers such as new technology, changing regulations, and shifting market demands change the nature of the business; this constantly shifts our perceptions about relative corporate strengths and weaknesses.

Uncertainties in price and market demand forecasts, and difficulties inherent in assembling the high-level asset performance information needed to accomplish these strategies, makes management of these portfolios a complex and demanding task. Data management strategies need to be structured to support the data query and analysis functions necessary to keep pace with the constant level of change.

Unlike centralized financial and human resource systems, information relating to reserves and production is dynamic and volatile. New information sources, such as those supplied by external vendors, consultants and GIS systems change rapidly and are rarely integrated into a form that can be shared and used throughout the organization. This information is often scattered in a variety of information 'silos', some centrally administered, some managed by departments, others are set up by individuals.

As a result, many, if not most, enterprises are faced with the following problems:

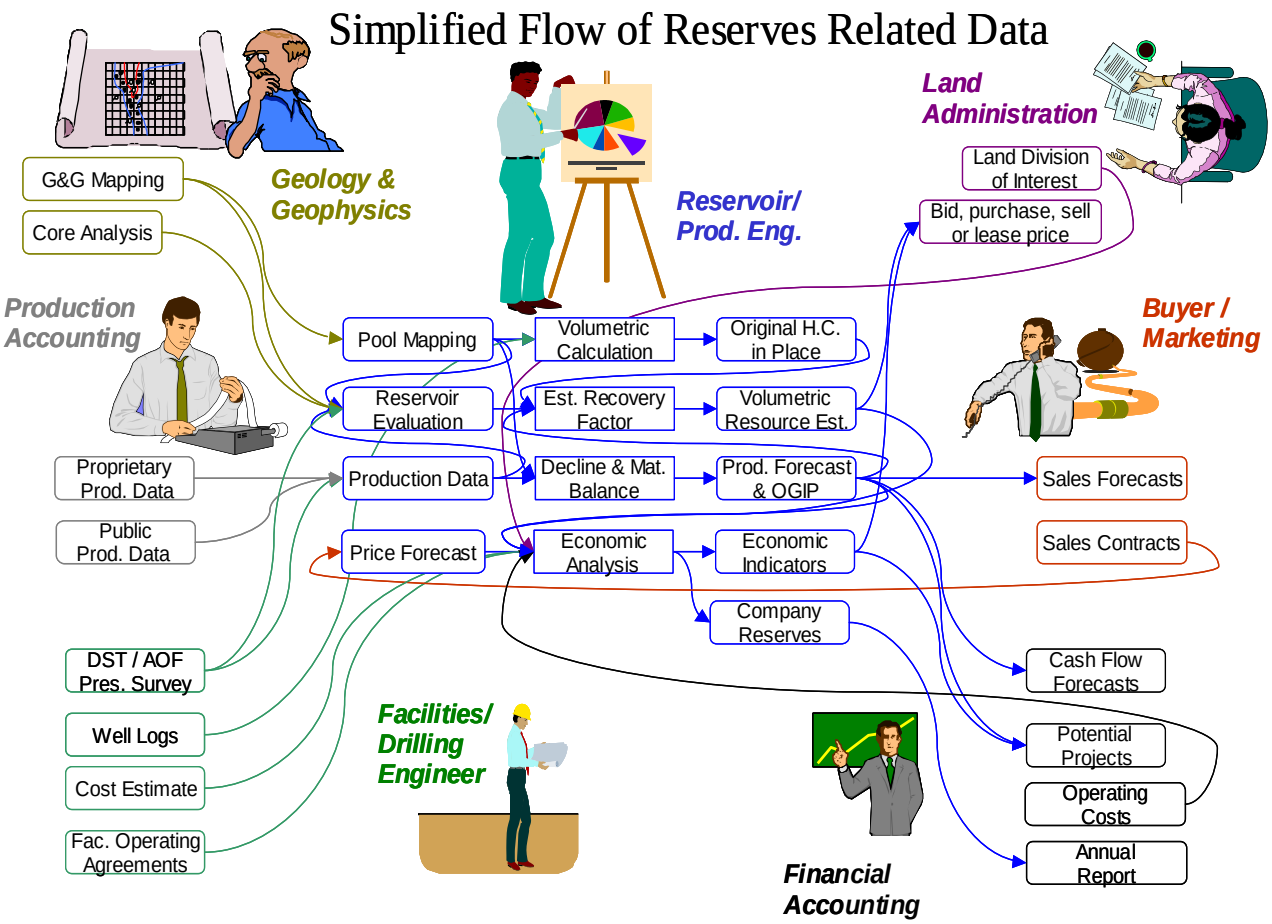
- Information is held in different formats and in different databases
- There is often inconsistency in the way the information is presented and analyzed between these sources
- It is almost impossible to integrate information from one data source with another. This can be because there is no common link (codes can be quite different between systems).
- Decision makers need to have a total, 'holistic' view of information

Integration of asset information and the corporate "portfolio" is critical to success in a data model, as is access to land, facilities, and proprietary well data necessary to see the full picture. Logical relationships between technical reserves and economic reserves must also be maintained. Models must provide the perspective to look at asset information the way the different business areas need to see it.

For a production engineer this means looking at the reserves and production associated with their area and facilities. For a Geologist, this means looking at proven, probable and possible reserves associated with their play. For a marketer, this means seeing production forecasts for proven reserves and approved capital projects, according to

product and sales points. Each perspective represents a twist on the accounting roll-up most often used to manage asset data. Each perspective is equally valuable in getting information to the person whose business decisions will impact the entire company.

Ironically, the lost opportunities in not having asset information readily available are not readily apparent - "you don't know what you don't know". To turn the picture around, however, even minor improvements in correlating business information across functional areas inevitably bring new opportunities to light.



Business Process Overview

Purpose

The Reserves Module provides a means of describing and managing information about Reserves data. This data is created throughout its life cycle, from the addition or acquisition of reserves, development and production of these reserves through disposition by sales or abandonment.

Description

The life cycle of Reserves data, from inception to final disposition through destruction or disposition, is lengthy. Reserves data is often processed and reprocessed many times in order to extract maximum value from the data; it is treated as a valuable asset in data sales and trades and provides important information for the design of new Reserves surveys. Each step in the life cycle is data intensive. Product generation and use of archived information at every stage mean that integration between Reserves data and a records management store is a critical component of effective Reserves data management.

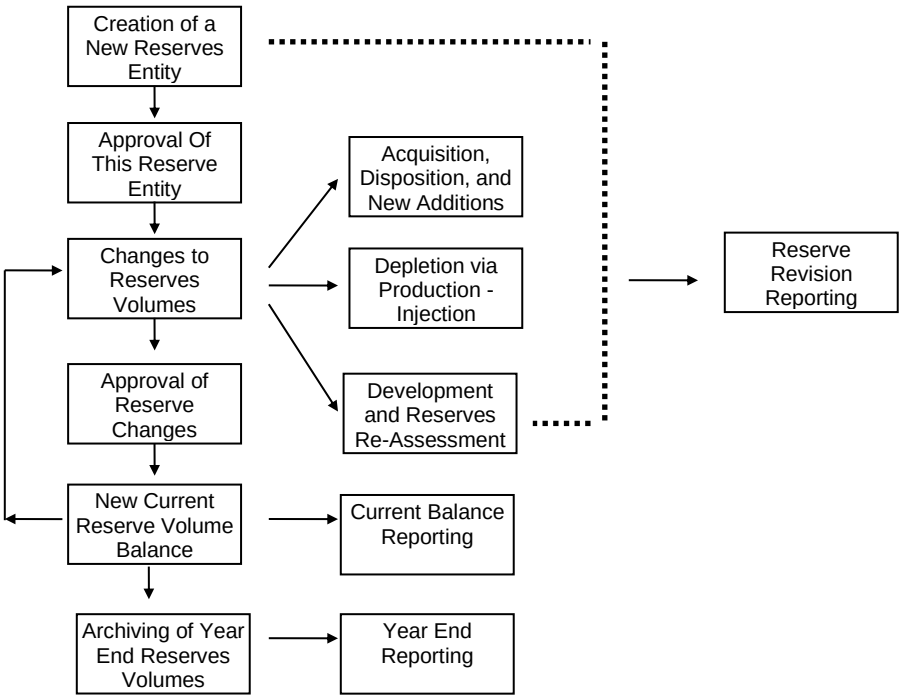


Figure 1: *The life cycle of a Reserves Entity. Good data management practices are critical to successful implementation of this workflow.*

Key Business Processes

Reserves Additions

A new reserve entity is generally created when a successful discovery well is drilled, when existing reserves are acquired from another company by purchase or trade, or when a productive zone is identified in an existing wellbore. With the exception of an acquisition, there is limited data available on which to make a reserves determination, or to attempt to forecast the production of those reserves over the life of the W. The initial reserves determination is usually made through volumetric calculations based on the aerial extent of the reservoir, the net pay thickness, porosity and water saturation of the reservoir rock. An alternative approach is to pick a similar well in the same field with an established reserves determination, and use these volumes by analogy. The initial production forecast for a new well is usually done by such an analogy.

Reserves Determinations

Volumetrics

The reserve volumes are calculated based on knowledge of the reservoir's physical characteristics as a container for the reserves. First the rock pore volume is determined by multiplying the aerial extent of the reservoir by its average net pay and average connected porosity. This yields the total volume available to contain moveable fluids. This is then multiplied by the fraction of the pore volume containing either gas or oil, which is usually calculated as 1 - Water Saturation and by a volume conversion factor between fluid volumes at surface conditions and fluid volumes at reservoir conditions. This yields the hydrocarbon volume originally in place. The last factor is the recovery factor, which is the percent of the total hydrocarbon in the ground that you expect to produce. This then results in the ultimate recoverable reserves volume.

Decline Analysis

As with any commodity being extracted from a container, the rate at which it comes out diminishes as it is being consumed (try this with ketchup for example). With oil and gas, this is usually because it is reservoir pressure which forces fluid to move from the reservoir to the wellbore, and as the fluid is produced the reservoir pressure is diminished. Decline analysis is a way of statistically analyzing production volumes which decrease with time and cumulative production under consistent operating practices, to arrive at a forecast which continues along this established trend to predict future performance. Eventually the forecasted production rate falls below a minimum threshold, and the well is assumed to be abandoned. If you significantly change the way you operate the well, you will change the rate at which it produces oil and gas. Such changes interfere with the validity of trying to perform decline analysis along the established trend.

This technique is usually considered the least ambiguous method of calculating remaining reserves, so long as a reasonable trend can be established, as it does

not rely on assumptions about the reservoir itself. The production volumes required for the analysis are usually readily available from independent data vendors, based on volumes reported to regulatory agencies. It is therefore the easiest analysis to perform on a well or field in which you do not have access to proprietary information.

Gas Material Balance

The basis of gas material balance is to assume that the reservoir acts like a closed tank of inert gas under pressure. As gas is removed from the reservoir, the pressure declines in a fashion consistent with the size of the tank and the compressibility of the gas itself. The underlying assumptions do not allow for movement of any fluid into the reservoir, phase change behaviour within the reservoir, or compaction of the reservoir rock due to settling, as the reservoir pressure diminishes. To perform the calculations, average reservoir pressures are required over the life of the reservoir. While a well is producing, the pressure immediately around the reservoir is lower than the average reservoir pressure. It is this pressure differential that causes gas to move toward the well bore. To achieve a measurement of average reservoir pressure, the well is shut in for a period of time to allow the pressure in the reservoir to equalize. Where the reservoir permeability is low, the time required to get true equalization may be many years, and as such, the pressure measurements made in these situations are only approximations. Generally though, the longer the well was left shut in before the pressure measurement was made, the more accurately the average pressure is determined. If more than one well is producing from the same reservoir, then ideally all the wells should be shut in while allowing the reservoir pressure to equalize. This loss of production is the most expensive component of acquiring accurate pressure data.

Reserves Approvals

Reserve volumes are a significant component of an E&P companies value as represented to shareholders, and are usually the major component. As such, accountability for changes to these volumes is important. Approvals may be based on both a technical assessment as well as on a business assessment. The technical approval covers off engineering verification that the estimated volumes are realistic for the certainty with which they are stated. The business approval covers off the funding commitments required to book undeveloped reserves as Proved – i.e. it is certain that the development of these reserves will happen in the future. Approvals may thus be required from multiple individuals before reserve changes are put in place.

Changes in Reserve Volumes

Once the appropriate approvals have been received, a mechanism is required to ensure that the effected balances changes are made. This is more an application issue than a business workflow.

Final Year End Numbers

At some point in the reserves reporting cycle, a decision is made to close off further changes to the reserve volumes for the current period (usually a year). The current balance at this time becomes the closing balance for that reserves reporting period, and forms the basis for subsequent corporate reporting. As reserve volumes may be subject to audit, it is required that the integrity of the detailed reserves information making up the reported reserve volumes is maintained.

Subsequent changes must be clearly identified with subsequent reserves reporting periods.

Production Adjustments

The most common reason for reserve volumes to change is that reserves are being produced. This is usually the one reason for reserves revisions which does not require any formal approval, and for this reason is often managed as an automated, or semi automated process.

Prior Period Production Adjustments

Occasionally, a reserves reporting period may be closed off with some of the production volumes for that period existing as an estimate, rather that an actual measured volume. There are also instances of production adjustments occurring that change the previous years "actual" production volumes. In both these situations a special reserve adjustment needs to occur when the correct production volumes become available. As the reserve numbers for the previous reporting period are not to be changed, the production revision must occur in the reporting period in which the discrepancy is identified. A separate revision category is usually used to separate such book keeping corrections from the production adjustments associated with the actual production for the current reporting period.

Processing and By Products

Oil as it exists in the reservoir usually contains dissolved gas. This effects its volume in situ. By the time it is measured at surface, however, most of the dissolved gas has separated out of the oil, and very little volume adjustments occur during further processing. Oil volumes are tracked at surface conditions, and though these volumes may be explicitly referred to as dead oil, or stock tank volumes, this is usually implied.

Gas, on the other hand, contains a number of impurities and by products that may be separated from the gas in multiple stages. The volumes measured for raw gas as it is produced at the wellhead may be different from the volumes of raw gas arriving at the plant inlet separator. Changes in pressure and temperature within the gas gathering system may have caused heavier hydrocarbons such as pentane to condense into a liquid usually called condensate. Within the gas plant undesirable impurities such as water vapour, Carbon Dioxide (if present in excessive amounts) and Hydrogen Sulphide (if present) are removed and

disposed of. Saleable by product such as Propane, Butane and C5 plus are removed both to ensure that the residual gas meets pipeline specifications, as well as to produce a saleable product.

An additional source of volume loss for gas is the consumption of part of the gas stream as fuel gas for running compressors and other production equipment.

To account for such loss in the overall gas volumes, loss factors are identified either for each processing stage, or for the overall volume loss between wellhead and sales meter. By product yield rates are also identified to establish anticipated natural gas liquids production resulting from gas processing.

To avoid confusion when reporting gas volumes, it is expected that raw and sale gas volumes be easily identified. A common convention is to report gross lease volumes on a raw gas basis, and partner shares on a sales gas basis.

Reserves Ownership and Reporting

Reserves reporting is usually required for both the total gross reserves (often called 8/8ths) as well as specific partner interest sets. In addition to identifying the partner's interest, the report should also identify the interest type being reported (Working Interest, Net Before Royalties, Net After Royalties, Revenue Interest etc.) Such requirements for reporting apply to both remaining volumes, as well as to those revisions which brought about the volume changes from the opening balance to the closing balance.

Reserves ownership arises from the mineral agreements in place for the land containing those reserves, though subsequent agreements for finding and producing the reserves may over ride the original mineral agreements. As such, there is no upper limit to the complexities which may be introduced into the ownership of the reserves. Even for a single well in a single zone, ownership may vary between the products being produced, and may also change over time. Ownership may even vary between the gas originally dissolved in the oil at time of discovery, and that which exists as a separate gas phase (or gas cap) immediately above the oil. As gas is gas when it comes out of the well, both engineering and accounting determination is required to establish what volumes are deemed to be solution gas, and what volumes are deemed to be gas cap gas.

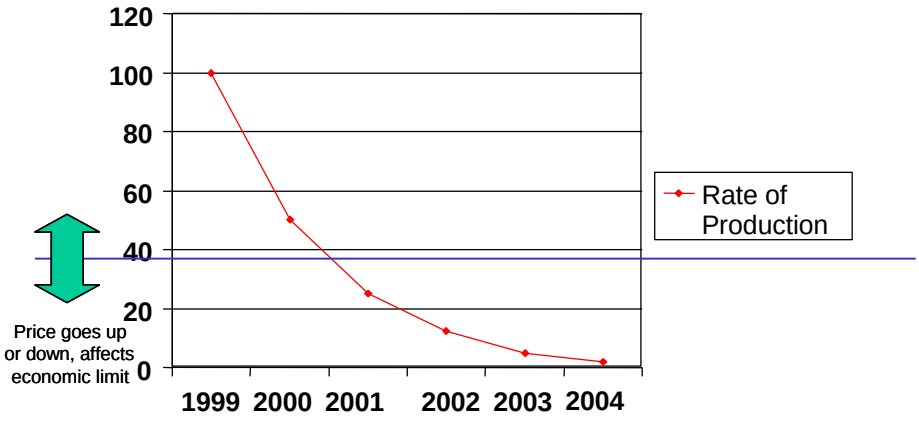
Revision Reasons/Categories

There are both business and corporate reporting requirements to categorize the reasons reserve volumes have changed. This is accomplished by defining a set of revision reason categories, and ensuring that when reserve revisions are made, the revision volume and associated category are also identified. Where a single reserve change contains components that apply to separate revision categories, then separate revision records are required for each component. Generally these categories group revisions by reasons such as: Drilling of new Exploration Wells, Drilling of offset wells, Acquisition or Disposition of properties, Interest changes due to payout, Capital Development, as well as production performance better or worse than expected.

Economic Evaluation

Reserve evaluations are usually made first on a technical basis – what the well is capable of producing, and then on an economic basis – what the well is capable of producing while maintaining a positive cash flow for the operator. Where the share of revenue and operating costs is not the same between owners, the point at which the cash flow becomes negative will be different for the different ownership positions.

Technical forecasts change only when the well's performance changes. Economic forecasts may change every time the price forecast fluctuates. At times of abrupt negative price changes, companies are faced with reporting reduced earnings as well as reduced reserves. The question usually asked is "Why should we write down the reserves, the oil/gas is still there isn't it?" The answer is "Yes, it's still there, but for this price regime more of it is going to be left there when the well becomes uneconomic."



Model Overview

Integration

Integration is the key to managing the Reserves Module and its components properly. Information critical to managing Reserves data throughout its life cycle is managed in many support and business modules in PPDM version 3.7:

Support Modules

Areas: business, regional or project areas associated with a Reserves set

Business Associates: track detailed information about partners, service providers and other people, companies and regulatory agencies that you do business with.

PPDM Unit Measure: captures the default stored unit of measure for any measured value in the database and conversion factors.

PPDM Volume Measure: capturers the fluid specific conversion factors and formulas which may be based on varying standard pressures and temperatures.

Entitlements: information about the rights that you have to any type of data and what you are able to do with it.

Facility, Field, Pool, Project, Strat Unit, Well: used to reference components which may make up a reserve entity, or to place a reserve entity on a reporting hierarchy.

Product: Validates the products for which reserves may be booked.

Business Modules

BA Interest Sets: describe partnership information for the ownership of Reserves volumes.

Contracts: contracts may be a component of a reserves entity.

Geodetic and spatial: use this module to reference any positional information to geodetic or cartographic information.

Land Rights: capture surface access rights associate with a Reserves entity.

Stratigraphy: make use of subsurface stratigraphic definitions that can be shared among all modules.

Obligations: used to manage obligations associated with reserve volumes – usually reporting obligations.

Projects: track work projects related to the development of reserves, or track reserve additions resulting from a specific project.

- Records Management: track the physical location of digital and hard copy products, circulation, retention, etc.

- Wells: describe in details wells that are part of the reserve entity.

Production Reserve Calculation

PDEN Decline: Captures forecast information resulting from decline analysis

PDEN Material Balance: captures the volumetric data and the results of gas material balance calculations, used to calculate reserve volumes

PDEN Volume Analysis: captures volumetric data and results, used to calculate reserve volumes

PDEN Volumes: captures the actual well production data used in decline analysis; material balance calculations; and to decrement reserve volumes based on historical production.

Contact PPDM to inquire about the status and availability of reference guides for these modules.

Data Diagrams

The diagram on this page is the legend for the tables discussed later in this document. Note that some or all of these elements may be present in data diagrams provided by the Association. Some elements are removed from final products to reduce file size:

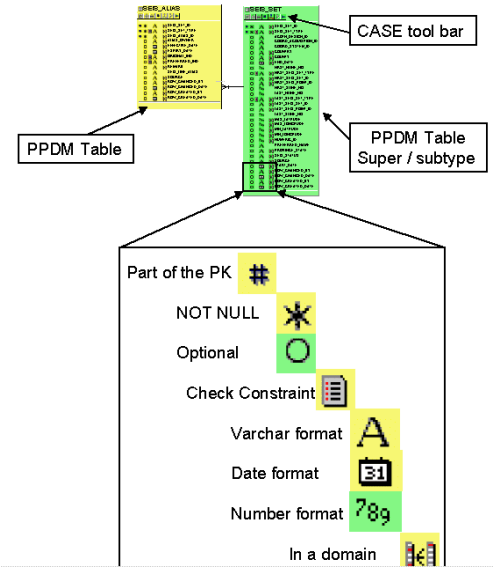


Figure 6: *This illustration shows the functions of each icon used in the data diagrams provided with PPDM version 3.7.*

The data diagrams for the Reserves Module diagrams are not provided in this reference guide because of their very large file size. Data diagrams can be obtained from the PPDM Association as part of the final model documentation or

as a set of PowerPoint diagrams. The PowerPoint diagrams will provide the best resolution for printed quality.

Tables and Columns: Reserves

The following tables exist in the Reserves module of PPDM version 3.7. Each table is described in the following section; you can jump to a table description by clicking on the hyper-linked table name below. Note that for detailed content descriptions for each table, you should refer to the PPDM version 3.7 table documentation.

Reserve Entities Set

Defines the Reserve entity and its components, and identifies the reserve classes associated with the reserves data. This set also holds the tables which define the reserve classes themselves, their origin, and the formulas that link them together.

Included in the RESERVES ENTITIES set is the RESENT_XREF table, which may be used to track the history of merging multiple reserve entities into a new entity, or fragmenting a reserve entity into multiple new entities.

Also included are the economic tables which identify the economic results associated with a reserve entity.

RESERVE_ENTITY

A reserve entity is the level at which reserves for a hydrocarbon deposit are determined, booked, and reported on. The reserve entity represents a grouping of one or more business objects that are usually unique to this reserve entity. The type of business object used to define the reserve entity is identified in the GROUP_TYPE column. These business objects are most often wells, pools, or fields. The business objects defining the reserve entity are stored in the RESENT_COMPONENT table, which may hold other business objects that are also associated with the reserve entity in some way. A hydrocarbon deposit has one primary product, usually oil or gas, just as a well has a primary product associated with it.

The RESERVE ENTITY table also identifies the most recent approvals, and most recent updates to the reserve entity.

Back to the list of table names

RESENT_COMPONENT

RESENT_COMPONENT is used to hold the list of business objects associated with a reserve entity. The particular set of business objects which define the reserve entity itself is defined in RS.GROUP_TYPE.

Also included in the RESENT_COMPONENT table are the REPORT_IND and PERCENT_CONTRIBUTION columns. These may be used to allocate reserve volumes back to one or more of the business objects that make up the reserve entity. If used, care must be taken that the sum of the PERCENT CONTRIBUTIONS for a reserve entity add up to 100% in order to ensure that allocated reserves do not over or under represent the actual reserve volumes for each entity.

Back to the list of table names

RESENT_XREF

The RESENT_XREF table allows associations to be defined between Reserve Entity ID's. Its intended use is to provide an audit trail when reserve volumes are moved from one reserve entity to another. This is expected to occur when multiple old entities are merged into one new entity, or a single old entity is allocated to multiple new entities. Though the XREF table can be used to group reserve entities up to a higher level, such as the Field level, it is more appropriate to use the reserve hierarchy tables to do so.

Back to the list of table names

RESENT_CLASS

The RESENT_CLASS table identifies the reserve class identifier(s) associated with a reserves entity. The reserve class is used to identify the confidence and status of the reserve volumes themselves. Where an implementation contains both Probabilistic and Deterministic reserve classes, there is a column in the RESENT_CLASS table which may be used to identify the confidence factor which links the methodologies of reserves reporting.

In an implementation where P10, P50 and P90 volumes are used to calculate Proved volumes, the CONFIDENCE_FACTOR may be used to identify the fraction of each probabilistic volume which is used to calculate the Proved volume. In an implementation where reserves are booked deterministically, the CONFIDENCE_FACTOR may be used to associate a probabilistic confidence level to the deterministic reserve classes. These confidence levels would be applied at the entity level. Reserve entities which make up a substantial portion of the booked reserves for a company would have Proved reserves booked at a fairly high confidence level (70% to 80%). Smaller reserve entities make a lesser contribution to total reserve volumes could be booked as Proved at a lower confidence level (60% to 70%).

Back to the list of table names

RESERVE_CLASS

The RESERVE_CLASS table identifies the specific confidence levels associated with a reserves class, as well as the production status and development indicator. The production status is generally PRODUCING or NON PRODUCING. The development indicator is generally DEVELOPED – most of the capital has been spent to bring the production to market, or UNDEVELOPED – a significant capital investment is still required to sell this production.

Back to the list of table names

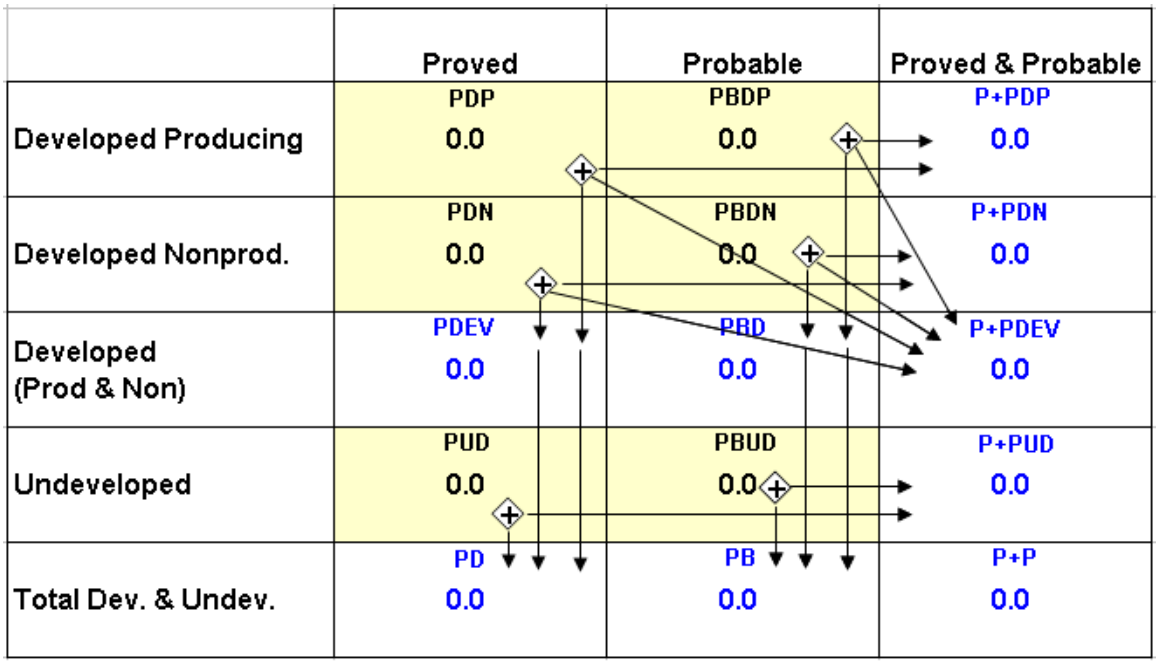
RESERVE_CLASS_FORMULA

The RESERVE_CLASS_FORMULA is used to define how other reserve classes are to be combined to yield a specific reserve class, and the specific conditions under which they are to be used.

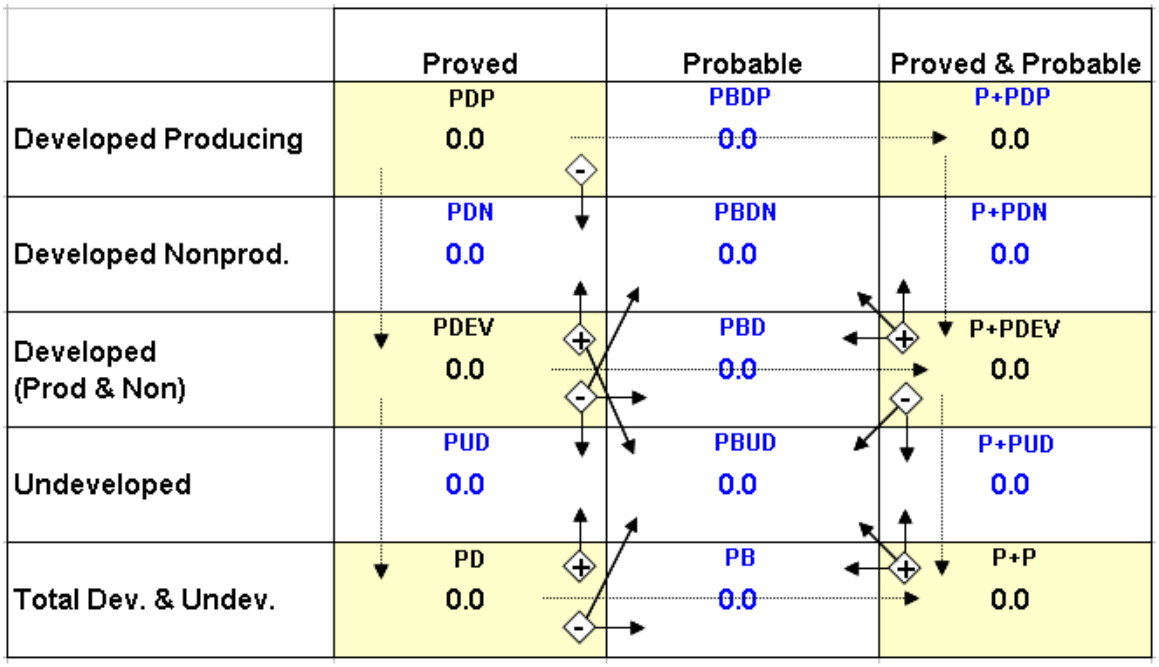
Generally speaking there are two separate methodologies for defining reserve classes. Partial development consists of identifying independent reserve classes which do not overlap. Partial development classes may include "Proved Developed", "Proved Undeveloped", and "Probable Undeveloped". To find the "Total Proved" volume it is always necessary to add up other reserve classes.

Full Development more closely follows the way reserves are established and developed, with successive categories also encompassing previous categories. Full development categories would include "Proved Developed", "Total Proved" and "Total Proved Plus Probable". To find the "Proved Undeveloped" volume one would subtract "Proved Developed" from "Total Proved". If there are only "Proved Developed" reserves booked for a given entity then the "Proved Undeveloped" volume is zero, and not the negative value of the "Proved Developed" reserves. In the same vein, if one were to report on "Total Proved" volumes, one would preferentially use the volumes booked under "Total Proved". If there are no volumes booked here then one would use the volumes booked for "Proved Developed" as they are a logical subset.

Partial Development sample data diagram



Full Development sample data diagram



Back to the list of table names

RESERVE_CLASS_CALC

This table contains the actual reserve classes to be used if a RESERVE CLASS FORMULA.USE RULE proves to be true, as well as the contribution factor to be applied. The sample data set and sample query assumes that the reserve volumes for a derived class is equal to the sum of all volumes associated with the RESERVE CLASS CALC.ORIGIN RESERVE CLASS ID multiplied by the RESERVE CLASS CALC.CONTRIBUTION FACTOR. To add the volume of an Origin Reserve Class the factor should be +1. To subtract the volume of an Origin Reserve Class the factor should be -1. For half of that volume the factor should be ½ . Other schemes are possible, however.

Back to the list of table names

RESENT_ECO_RUN

Use this table to identify the economic run and partner interest position associated with the values stored in the RESENT_ECO_SCHEDULE and RESENT_ECO_VOLUME tables. These tables allow the economic evaluation results to be associated with a reserve entity. Such an evaluation is assumed to be based on a specific set of price and operating cost assumptions which make up an economic scenario. The reserves model does not document these assumptions; it only identifies the economic scenario name. As multiple economic runs may be made under differing scenarios, for the same reserve

entity and reserve class, an ECONOMICS_RUN_ID is used to allow these records to co-exist. The value of a reserve entity is made up of the revenue of all the products being produced, less the cost of producing them, and any capital which must be invested.

Generally, economics are run on successive time periods until the total cash flow (revenue – expenses) goes negative. While in any given year it may be legitimate to invest more in capital equipment than the production is worth, the expectation is that there will be a return in future years. Once the value of the products produced is less than the cost to produce them, however, it's time to abandon the endeavor. The RESENT_ECO_RUN table identifies the number of years for which production volumes were scheduled (TECH_FORECAST) and the number of years for which these reserves generated a positive cash flow (RESERVE_LIFE_INDEX). The RESENT_ECO_RUN table also identifies the currency used in the RESENT_ECO_SCHEDULE table, as well as the Partner Business Associate and Interest Set.

Tax indicators (before or after) and any use of discounts rates is expected to be explicitly identified by the ECONOMIC_SCHEDULE used in the RESENT_ECO_SCHEDULE table. The detail of the discounting performed is also expected to be specified in the appropriate economic schedules.

Back to the list of table names

RESENT_ECO_SCHEDULE

This table holds the detailed schedules of economic results from an economic evaluation run. Schedules may include volumes, prices, costs, revenues or percentages. Schedule values may be specific to a particular date and may apply to a specific period, such as a month or a year. The column PERIOD_TYPE allows monthly, yearly, and summary values to readily identified. Schedules may identify key dates, such as payout or interest reversion. Schedules may also contain text descriptions identifying comments about the run, or textual information such as the discounting method used for discounted cash flows.

Each set of schedules is associated with a specific ECONOMICS_RUN_ID which reference a row in the RESENT_ECO_RUN table. This row identifies the Partner Business Associate for which the volumes and values apply, as well as the currency used and any currency conversion applied. The schedule types available are listed in the R_ECONOMIC_SCHEDULE table. There is a PRODUCT_TYPE column in the RESENT_ECO_SCHEDULE table which may used to identify the specific fluid in those instances where the SCHEDULE_TYPE might apply to more the one fluid (e.g. GROSSVOL). The PRODUCT_TYPE column would not be required if the implementation ensured that each SCHEDULE_TYPE was fluid specific (e.g. GROILVOL)

Tax indicators (before or after) and any use of discounts rates is expected to be explicitly identified by the ECONOMIC_SCHEDULE. The details of the discounting performed is also expected to be specified within this table.

Back to the list of table names

R_ECONOMIC_SCHEDULE

This is the validation table holding the list of allowable economic schedule types. In the sample data, the REMARKS column has been used to identify if a schedule type is fluid specific, if it may be fluid specific, or if it is not fluid specific. This is optional information only.

Back to the list of table names

RESENT_ECO_VOLUME

The individual economic volumes are stored in the RESENT ECO VOLUME table, and are usually less than the corresponding technical reserve volume stored in the RESENT VOL SUMMARY table. It is good practice to ensure that the products identified in the economic analysis match the products identified in the RESENT_PRODUCT table, though this is not enforced by the model.

Back to the list of table names

Reserve Summaries Set

Defines the products which make up the reserve entity, the properties of those products, and the volumes associated with the reserve entity. Both the balance summary volumes, and volume revisions to those summary volumes are stored, as well as partner information which links the volumes back to ownership information.

RESENT_PRODUCT

Defines the products associated with a particular reserve entity and reserve class. In most situations where a reserve entity has multiple reserve classes the products will be the same between reserve classes. There are a few exceptions, generally when future development is anticipated to change the products recovered. The model is set up to support such exceptions.

Back to the list of table names

RESENT_PROD_PROPERTY

Specifies the detailed product properties specific to a reserve entity and reserve class. These properties are used for some of the volume conversions. Where gas volumes are converted into a heating value equivalent the HEAT_CONTENT is used. When oil is converted into a weight equivalent the OIL_DENSITY is used.

The LOSS_FACTOR is used to define the processing loss from one related product type to another, such as from RAW GAS to SALES GAS.

Back to the list of table names

PRODUCT

Previously R_PRODUCT_TYPE, this table is now more than merely a reference table for the RESENT_PRODUCT and PRODUCT_COMP tables. It now stores the CONVERSION_QUANTITY associated with each defined product type. This allows a single set of conversion factors to be defined for GAS_VOL and OIL_VOL for each unit regime. GAS_VOL is then the CONVERSION QUANTITY used for all of the various gas product types (Raw Gas, SLN Gas, Sales Gas etc.)

The Product table also stores the preferred unit of measure associated with each product. This supports the data management principle of using the same units of measure for the same product type and measurement type, across the entire database.

Back to the list of table names

PRODUCT_COMP

Commonly, a reserves product is comprised of other reserves products. For example, Natural Gas Liquids may be defined as the sum of Propane, Butane, and Condensate. The PRODUCT_COMP table allows for definition of these sub products to allow roll up reporting at a summary product level. Other products that may be defined are Oil Equivalent and Gas Equivalent. This allows a single reserve volume to be calculated for each reserve entity and class, and may be used to rank reserve entities which contain differing product mixes.

Back to the list of table names

RESENT_VOL_SUMMARY

This table identifies the opening and current balance for each product associated with a reserves entity and reserves class. All records for a summary (such as the annual summary) should use the same SUMMARY ID.

In addition to the Gross volumes (GROSS_SUMMARY_IND = 'Y'), net interests may also be defined for each INTEREST SET PARTNERS (NET_ SUMMARY_IND = 'Y'). Separate records for each ownership position should be identified by a unique SUMMARY_OBS_NO. Generally, the gross summary would be set with OBS_NO = 1, and partner interests using OBS_NO = 2…n.

The RESENT_VOL_SUMMARY table also tracks the method used to establish the reserve volumes for each product, and contains foreign keys linking to records in the PDEN DECLINE, PDEN VOLUME ANALYSIS and PDEN MATERIAL BAL tables. This allows you to associate your results with the technical analysis that was used as input for creating the summary volumes.

If the product is a by product of another product in that reserve entity, then the YIELD PARENT PRODUCT, and YIELD_RATE may be specified. This yield rate may be derived from the RPP.LOSS_FACTOR where Sales Gas is a by product of Raw Gas.

Industry practice is inconsistent in the way by products are calculated. Both Sales Gas and Raw Gas volumes are currently used in the denominator when the by product yield is calculated. It is strongly recommended that within a specific implementation a consistent approach be maintained.

Finally, the RESENT_VOL_SUMMARY table identifies the creation and approval date for the volume summary record, the approving analyst ID, and the effective date and expiry date for the record itself.

Back to the list of table names

RESENT_VOL_REVISION

Tracking changes to reserve volumes is almost as important as tracking the resulting volume balances. There are business and reporting requirements to identify not only the revision date and approvals received, but also the underlying business process which led to the reserve changes. The business process is often referred to as a revision reason or revision category. It is used to support reporting of all the reserve changes associated with the drilling of new exploration wells, production from existing reserve entities, or the acquisition and disposition of reserves through sale or trade.

The RESENT VOL REVISION table links directly to the RESENT VOL SUMMARY table. Multiple revision records may be associated with a single volume summary record.

The RESENT VOL REVISION table tracks its own approval information, which may be independent of the vol summary approvals. Also tracked is a project ID, which allows multiple revision records in multiple reserve entities to be associated with the same Project ID. This is useful where large drilling and development projects result in changes to multiple reserve entities.

Back to the list of table names

RESENT_REVISION_CAT

This table tracks revision categories set up to identify the business processes associated with reserve revisions. Each revision category is part of a category type, which may be used to roll up differing types of revisions into subsets required for reporting purposes, such as Revisions, Adjustments, and Acquisitions. Categories may also be defined as part of other categories, for roll up and reporting purposes.

Back to the list of table names

Reserve Hierarchy Set

This group of tables places business objects (PDEN's or Reserve Entities), within a corporate hierarchy which may be used for roll up reporting, grouping or display purposes. Two types of hierarchies are defined. Generalized hierarchies are created to define the types of hierarchies that are authorized for use (COUNTRY, PROVINCE, DISTRICT). Specific hierarchies are created to implement and support reporting roll-ups as shown below.

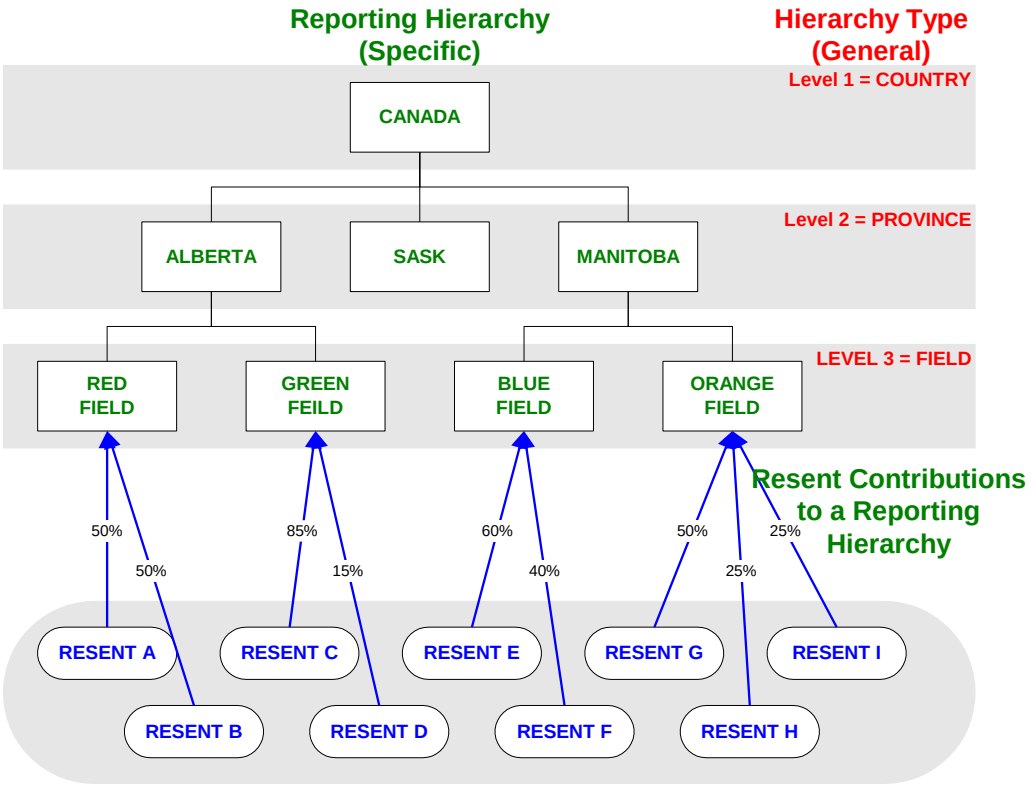


Figure 3: Shows the relationship between generalized and reporting hierarchies and how individual reserves entities may contribute to a reporting hierarchy.

RESERVE_HIER_TYPE

This table tracks generalized hierarchy classifications or types (such as a geographic hierarchy). Each type of hierarchy may be given a name, such as Geographic or Organizational. Details about the levels permissible in this type of hierarchy are defined in RSHD.

Back to the list of table names

RESENT_HIER_DESC

This table tracks the levels that are permitted in a generalized hierarchy that is defined in RESENT_HIER_TYPE, such as COUNTRY, PROVINCE and FIELD in the example in Figure 3. The level in the hierarchy is indicated with

LEVEL_SEQ_NO. Each level may only be reported once, and may have only one type of business object associated with it (LEVEL_TYPE).

Back to the list of table names

RESENT_REPORT_HIER

These are actual reporting hierarchies that have been set up to support reserve rollup reporting. Differing business functions, and diverse reporting requirements often necessitate more than one corporate hierarchy be set up to track reserves and reserves data. For Geology and Geophysical staff, useful reporting may be at a Field and strat unit level. Accountants may want to see reserves reported at business unit and cost center levels. Production staff may want to see reserves analysis done at a facility level, where their business decisions are made.

This model supports setting up multiple hierarchies, each with its own name and type. A reporting hierarchy might be called the Western Canada Reporting Hierarchy, or the Northern Division Reporting Hierarchy.

Back to the list of table names

RESENT_HIER_LEVEL

This table may be used to associate a specific level in a hierarchy with the specific business objects that are used to create the reporting hierarchy.

In the example above, a reporting hierarchy based on the GEOGRAPHIC TYPE may have COUNTRY defined as level 1 of the hierarchy (captured in RESERVE_HIER_DESC). The table RESENT_HIER_LEVEL shows that for the CANADA reporting hierarchy, CANADA is used as the first level. At the second level (PROVINCE) we are using Alberta, Saskatchewan and Manitoba and at the third level (FIELD) we are using the RED, GREEN, BLUE and ORANGE fields.

A recursive relationship in this table allows you to explicitly indicate the parentage of each object in the hierarchy (the RED field is in ALBERTA which is in CANADA).

There are foreign keys to the AREA, CONTRACT, LAND_RIGHT, FIELD, POOL and STRAT_UNIT tables. It is anticipated that additional foreign keys will be added for any implementation which requires additional hierarchy types.

Back to the list of table names

RESENT_HIER_USE

This table identifies the PDEN ID's and RESENT ID's of the business objects associated with a specific hierarchy level. Usually, you would only associate RESENTS or PDENS at the lowest level of the hierarchy, as has been done in the example. If you wish, however, you may explicitly roll up the resents into all levels of the hierarchy.

Back to the list of table names

Production Reserves Forecasts Set

These tables hold the Decline Case and Decline Segment information associated with Decline Analysis, as well as data derived from material balance and volumetric analysis.

PDEN_DECLINE_CASE

As with any commodity being extracted from a container, the rate at which it comes out diminishes as it is being consumed (try this with ketchup for example). With oil and gas, this is usually because it is reservoir pressure which forces fluid to move from the reservoir to the wellbore, and as the fluid is produced the reservoir pressure is diminished. Decline analysis is a way of statistically analyzing production volumes which decrease with time and cumulative production under consistent operating practices, to arrive at a forecast which continues along this established trend to predict future performance. Eventually the forecasted production rate falls below a minimum threshold, and the well is assumed to be abandoned. If you significantly change the way you operate the well, you will change the rate at which it produces oil and gas. Such changes interfere with the validity of trying to perform decline analysis along the established trend.

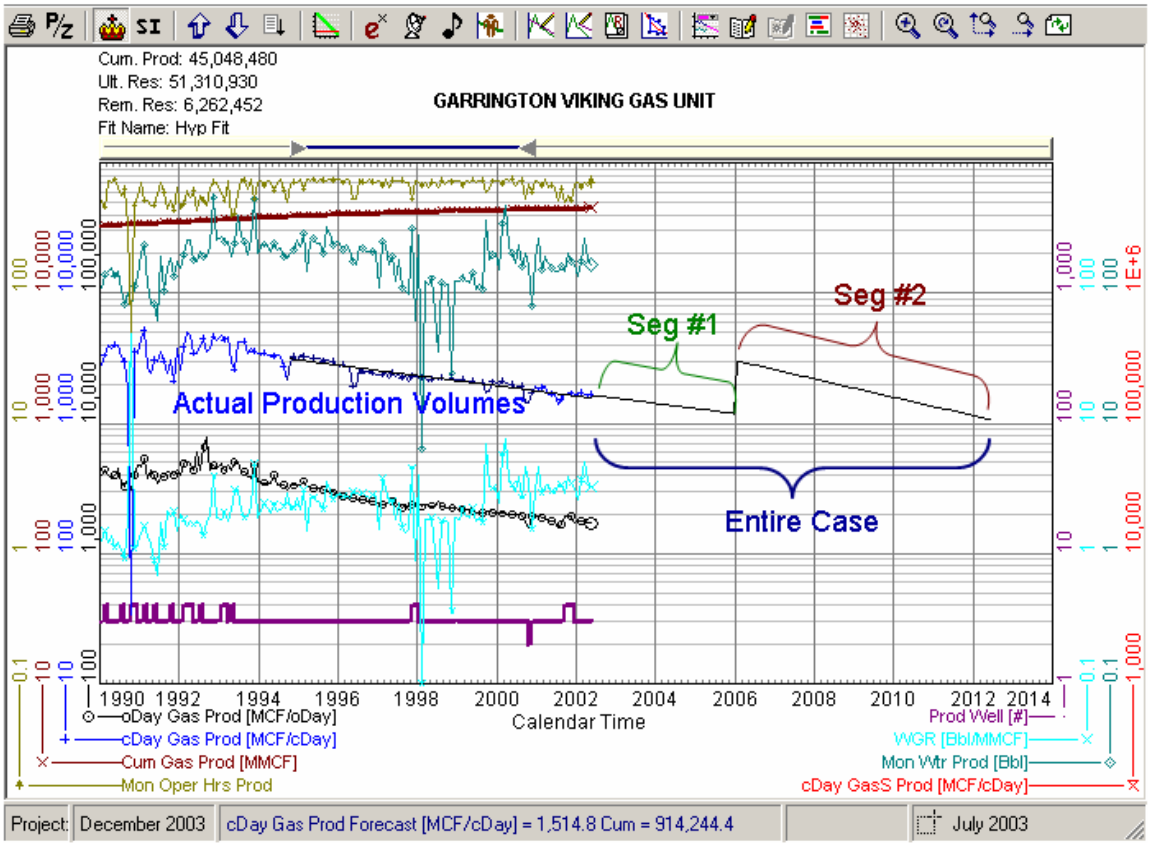
The PDEN_DECLINE_CASE table holds the case level summary information associated with an entire Decline Analysis forecast. The forecast itself may be made up of one or more segments. Generally a segment covers a portion of the decline forecast based on a single set of decline curve parameters, such as initial rate, decline rate, and exponent. Where the forecast shifts abruptly due to planned changes in operating conditions (e.g. installation of additional compression facilities) a second decline segment is used.

Back to the list of table names

PDEN_DECLINE_SEGMENT

This table holds the individual segment parameter sets, along with the summary data for each segment.

See sample Decline analysis graph below:



Back to the list of table names

PDEN_DECLINE_CONDITION

This table holds any decline data that does not fit in the Case or Segment tables. It can be used to hold non forecast related data such as well count and service factors. It can also be used to hold monthly forecast volumes associated with the production forecasts from the Segment table to make this data directly available for economic evaluation.

Back to the list of table names

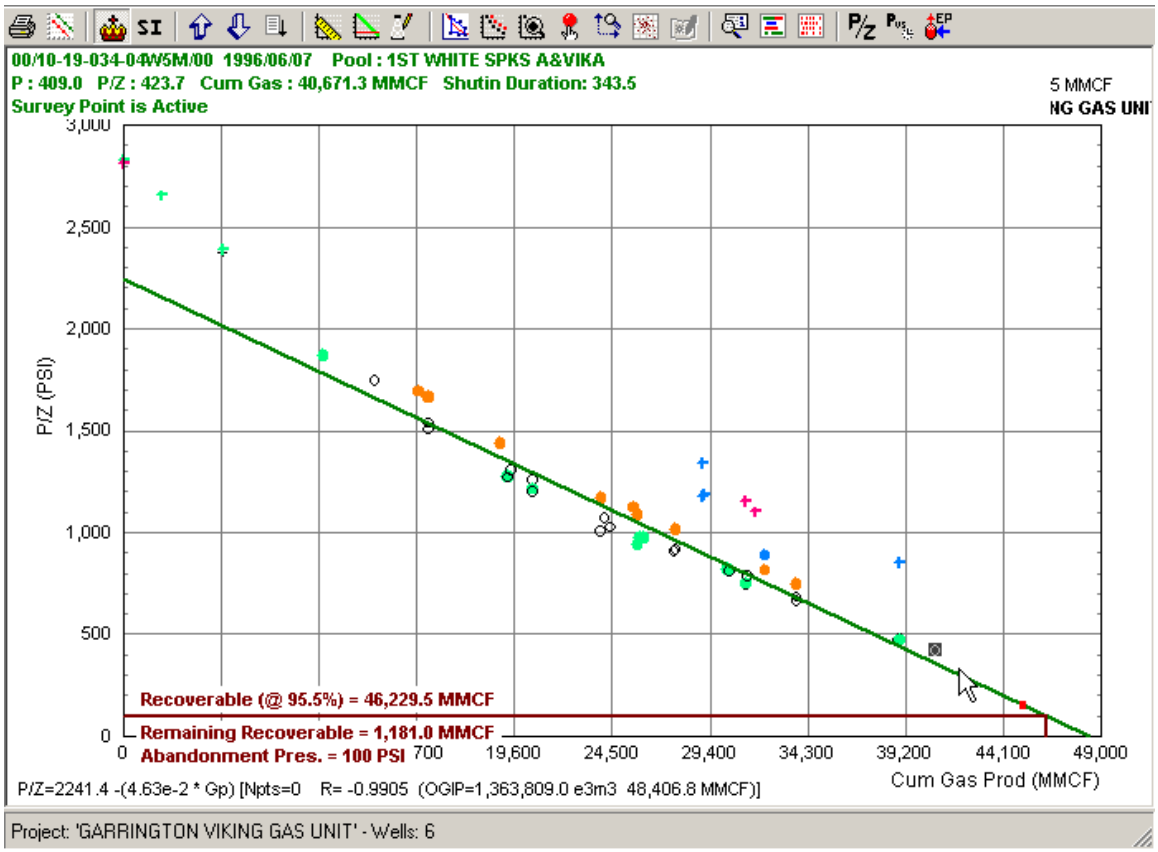
PDEN_MATERIAL BAL

The basis of gas material balance is to assume that the reservoir acts like a closed tank of inert gas under pressure. As gas is removed from the reservoir, the pressure declines in a fashion consistent with the size of the tank and the compressibility of the gas itself. The underlying assumptions do not allow for movement of any fluid into the reservoir, phase change behaviour within the reservoir, or compaction of the reservoir rock due to settling, as the reservoir pressure diminishes.

To perform the calculations, average reservoir pressures are required over the life of the reservoir. While a well is producing, the pressure immediately around the

reservoir is lower than the average reservoir pressure. It is this pressure differential that causes gas to move toward the well bore. To achieve a measurement of average reservoir pressure, the well is shut in for a period of time to allow the pressure in the reservoir to equalize. Where the reservoir permeability is low, the time required to get true equalization may be many years, and as such, the pressure measurements made in these situations are only approximations. Generally though, the longer the well was left shut in before the pressure measurement was made the more accurately the average pressure is determined. If more than one well is producing from the same reservoir, then ideally all the wells should be shut in while allowing the reservoir pressure to equalize. This loss of production is the most expensive component to acquiring accurate pressure data.

The PDEN_MATERIAL_BAL table holds the generic fluid property, cumulative production volumes and overall recovery factors that are used for Material Balance Calculations. The table also holds the summarized results from a gas material balance.



Back to the list of table names

PDEN_VOLUME_ANALYSIS

With volumetric analysis, the reserve volumes are calculated based on knowledge of the reservoir's physical characteristics as a container for the reserves. First the rock pore volume is determined by multiplying the aerial extent of the reservoir by its average net pay and average connected porosity. This yields the total volume available to contain moveable fluids. This is then multiplied by the fraction of the pore volume containing either gas or oil, which is usually calculated as 1 - Water Saturation and to a volume conversion factor between fluid volumes at surface conditions and fluid volumes at reservoir conditions. This yields the hydrocarbon volume originally in place. The last factor is the recovery factor, which is the percent of the total hydrocarbon in the ground that you expect to produce. This then results in the ultimate recoverable reserves volume.

The PDEN_VOLUME_ANALYSIS table holds the reservoir and fluid property data used to perform volumetric reservoir calculations, as well as the reserve volumes resulting from these calculations.

Production Volume Regime Set

PPDM_VOL_MEAS_REGIME

When reserve determinations cross regulatory jurisdictions, it is often necessary to adjust the conversion factors, especially those used for gas volume conversions. Gas is by its nature compressible, and it will expand or contract as pressure and temperature change. When a cubic foot or cubic meter of gas is measured, it is necessary to know the standard pressure and temperature at which the volume was measured. From a practical perspective the effect of these changes is minor indeed, however, when large reserve volumes are being managed, even a difference of a fraction of a percent in the conversion factor can result in significant volume discrepancies.

The RESENT_VOL_REGIME table allows for a number of volume measurement regimes to be defined, each with its own standards for gas and oil measurement.

Back to the list of table names

PPDM_VOL_MEAS_USE

This table identifies the political and geographical regions for which a specific measurement regime would apply. In Canada a single measurement regime is adequate, as standard conditions have been defined federally. In the United States, multiple measurement regimes may be needed, as measurement standards have been defined at both the state and federal levels.

Back to the list of table names

PPDM_VOL_MEAS_CONV

This table identifies the actual conversion factor, and formula that may be required for converting volumes for each defined CONVERSION QUANTITY. Conversion quantities would be such measurement types as Gas Volumes, or Oil Volumes. The use of a Conversion Quantity, rather than using the product type directly, avoids the need to set up conversion equivalents for each type of gas, oil, or hydrocarbon fluid that may exist in the PRODUCT table.

The conversion factor is the multiplier to transform the data value from the FROM_UOM to the TO_UOM. The formula includes any other data that may also be used as part of this calculation. Where gas volumes are converted into a heating value equivalent the HEAT_CONTENT is used. When oil is converted into a weight equivalent the OIL_DENSITY is used. The LOSS_FACTOR is used to define the processing loss from one related product type to another, such as from RAW GAS to SALES GAS.

Back to the list of table names

RESENT_VOL_REGIME

This table explicitly defines the volume measurement regime to be used for a given Reserve Entity. Note that the regime to be used may change over time.

Back to the list of table names

PDEN_VOL_REGIME

This table explicitly defines the volume measurement regime to be used for a given PDEN Entity. Note that the regime to be used may change over time.

Back to the list of table names

Production Reporting Subtypes (New Tables)

PDEN_RESENT

This table allows a reserve entity to be a valid subtype of the PDEN table, without consideration to the reserve classes defined for that entity. PDEN_RESENT should be used when tracking actual production volumes for a reserves entity. Projected or forecast volumes are tracked using PDEN_RSC.

Back to the list of table names

PDEN_RESENT_CLASS

This table allows a reserve entity and reserve classification to be a valid subtype of the PDEN table, . This table should be used when tracking projected production volumes for a reserves entity. Real volumes are tracked using PDEN_RESENT.

Back to the list of table names

Implementation Considerations

Constraints in PPDM

It is essential that anyone who is considering using PPDM version 3.7 review the Constraints Reference Guide first. Improper use or population of constrained columns in PPDM can compromise the quality of your data and the reliability of your queries. This document may be obtained from the PPDM Association or downloaded from the PPDM web site at www.ppdm.org.

Check Constraints

PPDM version 3.7 makes use of check constraints in rare cases where the values that may be input for a column are known at design time and will not change over time. Two types of uses are observed in PPDM 3.7.

- Where the column name is %_IND, the column is an indicator field, and the values may only be Y, N, or null.
- Super-sub type implementations use check constraints to enforce the integrity of the super-sub type relationship. Currently these relationships are in use for, Seismic Sets, Records Management, Support Facilities, Production Entities and Land Rights.

  Let's use Seismic Sets as an example. This structure consists of a parent table (SEIS_SET) and eight sub-type tables (SEIS_3D, SEIS_ACQTN_SURVEY, SEIS_INTERP_SET, SEIS_LINE, SEIS_PROC_SET, SEIS_SEGMENT, SEIS_SET_PLAN and SEIS_WELL). Each of the tables has a two-part primary key: SEIS_SET_ID and SEIS_SET_TYPE.

  SEIS_SET_ID is assigned by the user and can have any value as long as it is unique for that type of Reserves set. SEIS_SET_TYPE was designed to maintain the integrity of the super-sub type structure and can only have the values assigned to it by check constraints; these values are the table names of the eight valid sub-types. In SEIS_SET, the SEIS_SET _TYPE can have any of the table names, but in each of the sub-types, it can only have the name of the table it is owned by.

Currencies in PPDM

Costs in PPDM may originate in any valid Unit of Measure (UOM), such as USD, $CDN, YEN, etc. However, to ensure that queries for retrieval and reporting are efficient, it is desirable to convert all original currencies to a standard unit of measure for storage in the database. PPDM supports the requirement to restore the original value in the following way:

- Convert all stored currencies to a single currency type, such as US dollars.

- CURRENCY_OUOM stores the currency in which the funds were initially received. When the stored currency is multiplied by the CURRENCY_CONVERSION, the value of the transaction in the original currency is obtained.

- CURRENCY_CONVERSION stores the rate applied to convert the currency to its original monetary UOM from the stored UOM. This value is valid for this row in this table at the time of conversion only. When this value is multiplied by the stored currency value, the original value of the transaction in the original currency is restored.

Units of Measure

Relational databases, powerful as they are, are not good at certain types of query and retrieval. Any time a query is developed that requires the database to retrieve all the rows in a large table and perform some calculations on the data before returning results to a user is likely to perform very poorly. This assumes, of course, that the person constructing the query is aware that a calculation is necessary when writing the query. Data management strategies for such tables recommend that requirements for on-line conversions such as this be eliminated if at all possible. The PPDM strategy for handling units of measure falls into this category.

Every column in the data model that references a Unit of Measure (such as a depth, temperature, length etc.) should be stored using a single, common unit of measure. For example, in one PPDM instance, all the total well depths should be stores as meters or as feet. Storing some depths as meters and the rest as feet creates problems for the data base and adds confusion to the user (who may not be aware that the numbers in the DEPTH column are not all meters).

The original unit of measure (the unit in which the data was originally received) can be stored in the data table. For example, the WELL table captures FINAL_TD and FINAL_TD_OUOM. These columns capture the value of the final total depth of the well and the units that the depth was originally captured in.

The ***stored unit of measure*** is captured in the PPDM meta model, PPDM_COLUMN. This table captures the default unit of measure for a column and the name of the column where the original unit of measure is stored. The following illustration provides an example:

WELL

UWI	DRILL_TD	DRILL_TD_OUOM
SMITH12F	1250	FEET
JONES44	1560	METERS
12345	1400	FEET

PPDM_COLUMN

TABLE_ NAME	COLUMN_ NAME	UOM_ COLUMN	OUOM_ COLUMN	DEFAULT _ OUM_ SYMBOL
WELL	UWI			
WELL	DRILL_TD		DRILL_TD_OUOM	M
WELL	DRILL_TD_OUOM			
WELL_CEMENT	CEMENT_AMOUNT	CEMENT_AMOUNT_UOM	CEMENT_AMOUNT_OUOM	

Figure 9: *The method for storing and tracking units of measure is illustrated here..*

Note that in the example, the Drilling TD is stored in meters, but was originally received as feet. In some cases, it is not possible to ensure that all the rows in a column are stored as a single unit of measure – this is common in cases where the unit of measure is dependent on some other factor. For example, substance measurements may depend on the substance being measured; gases are stored as MCF, liquids as BBL etc. In these cases, the unit of measure is stored directly in the business table.

Audit Columns

Each table contains five columns: SOURCE, ROW_CHANGED_BY, ROW_CHANGED_DATE, ROW_CREATED_BY, and ROW_CHANGED_DATE. These columns satisfy a data-auditing requirement to identify the user and date of database transactions.

Use the “CREATED” columns when you are inserting new data rows and the “CHANGED” columns when you are updating a data row. The ROW_CHANGED / CREATED_BY columns are usually populated using the system login id in use. ROW_CHANGED / CREATED_DATE is usually set to the system date of the insert or update operation.

To populate the SOURCE column, specify where you obtained the data. If you receive the data from Vendor A, and Vendor A received the data from Regulatory B, you should set the SOURCE to Vendor A. In some cases (such as for interpreted picks), data is created by an application. In this case, the source may be set to identify the application that created the data.

Identifying Rows Of Data That Are Active

Maintaining information about how a business object has changed over time is an important business requirement for all these modules. To support this, mechanisms for allowing versioning have been added to many tables.

Many tables in PPDM version 3.7 contain a column called ACTIVE_IND. The values for this column may be one of Y, N, or null. When more than one row of data (such as a spatial description or a status) has been created for a business object, use the ACTIVE_IND to indicate which row is currently active (note that in some cases, more than one row may be active simultaneously).

This provides implementers with two benefits. First, when populating EFFECTIVE_DATE and EXPIRY_DATE it will not be necessary to populate EXPIRY_DATE with a false future date to indicate that the row of data has not expired yet. Second, queries can explicitly search only for rows that are active.

If this column is used for queries, as recommended (such as "find me the currently active status for this land right"), you should implement procedures to ensure that this column is always populated as either Y or N and maintained appropriately. If the column is left blank (NULL), the query will not be consistent or reliable.

For example, you could default the value to N if the expiry date is filled in and has already happened. Make it Y if the expiry date is empty *or* if the expiry date contains a future date.

Modifying the Reserves Module

Subsetting PPDM

The PPDM data model is designed to allow users to implement portions that support their business without needing to manage modules that are not required. Good data management practices are also supported; this means that data redundancy is reduced in the Model whenever possible.

All information about Reserves will be found in the Reserves module; information about contracts is stored in the Contracts module, details about objects that are retained for long term use are stored in the Records Management module and so on. Depending on your business requirements, you can implement all or some of the modules.

In general, it is usually simplest to install the entire PPDM data model and simply restrict usage to the portions that are useful to you. Additional tables can be implemented as your business requirements expand, or as your data and processes are able to support capture in a data model. Architectural guidelines for subsetting PPDM are contained in the PPDM Architectural Principles Document. This document can be obtained from the PPDM Association or downloaded from the PPDM web site at www.ppdm.org.

Expanding PPDM

As a consequence of the PPDM Design process, which actively solicits and incorporates business requirements from Industry, many users find that the model is quite complete. However, individual implementations may find that additional columns are needed, or that some denormalization will help their performance.

The Association provides documentation about how to expand the data model to accommodate your specific requirements. This document can be obtained from the PPDM Association or downloaded from the PPDM web site at www.ppdm.org.

Feedback to PPDM

Much of the growth of the PPDM model can be attributed to Industry feedback. All implementers are requested and encouraged to provide feedback to the Association about changes they have made for implementation. Feedback can be submitted to changes@ppdm.org.

Frequently Asked Questions (FAQ)

Which is primary product for a reserve entity?

The primary product is identified by RESERVE ENTITY.PRIMARY PRODUCT TYPE

What are my total oil reserves, including heavy oil, sour oil, light and medium oil?

The PRODUCT COMP TABLE allows the definition of Heavy, Medium and Light oil, as well as the definition of Total Oil. the PRODUCT_COMP table would then be used to bring them together (Total Oil = Heavy Oil + Medium Oil + Light Oil etc.).

Is prior year data available?

Reserves estimates and production volumes can all be tracked over time. In the RESERVE Entity table the column ROW CREATED DATE identifies when reserves are added. The EFFECTIVE and EXPIRY columns in the RESENT VOL SUMMARY and RESENT REVISION VOL tables identify the records associated with each time period.

How long will my reserves last me?

The RESENT VALUE .RESERVE LIFE INDEX will hold the reserve life index calculated by the economics program. Like the other economic indicators the reserves model does not track how the economic calculations or indicators were established.

See RESENT VALUE. TECHNICAL FORECAST for the technical forecast life, or refer to PDENDFC.END DATE.

What is the basis for the current reserve estimates?

The RESENT VOL SUMMARY table has a column Volume Method which identifies the methodology used to establish the reserve volumes for a specific reserve entity and reserve class ID. Foreign keys are available to link the volume

summary to a specific Decline Analysis Case ID, Material Balance ID, and Volume Analysis ID.

How do I track monthly forecasted volumes for a reserve class?

Monthly data can be entered into the table PDEN DECLINE SEGMENT, though this is not the ideal use for this table.

Preferably, you should create a PDEN_RESENT_CLASS to track forecast volumes by reserves class. PDEN volume summary tables can be used to store product volumes associated with an activity type such as "FORECAST".

The third and least desirable option is to use the PDEN DECLINE CONDITION to store the monthly forecast volumes.

How can I keep track the well count associated with a forecast?

If the PDEN VOL SUMMARY table is used to hold the forecast, columns are already available for well counts. Another option is to use the PDEN DECLINE CONDITION table to hold well count or other data.

How do I handle multiple types of gas shrinkage?

Each level of process at which reserve volumes are tracked can be created as its own product type. A database can contain Raw Gas, Separator Gas, and Sales Gas. Specific loss streams may also be identified such as Fuel Gas, and Flared Gas. The RESENT_PROD_PROPERTY table contains a column LOSS_FACTOR for specifically identifying the expected loss from one level of processing to the next. Loss streams may also be defined as byproducts of a parent product.

Fuel Gas may be defined as a byproduct of Raw Gas in the RESENT VOL SUMMARY table, with a yield rate of a couple of percent. The PRODUCT_COMP table also allows products to be defined in relation to other products. In this fashion Sales Gas may be defined as Raw Gas minus Fuel Gas.

How do I roll reserve volumes up to corporate reporting levels such as Field and A.

The RESENT_HIER_USE table identifies all the reserve entities associated with a given position on a specific Reserve Hierarchy. Volumes for these entities may then be summed to arrive at the corporate totals for these areas.

How is the reserves estimate divided among various levels of description?

RESENT COMPONENT.PERCENT CONTRIBUTION identifies the allocation factors back to individual components, or to other reserve entities.

Which wells, production strings, completions are included in this reserves entity?

The table RESENT COMPONENT tracks all the business objects associated with a specific reserve entity.

What are the economic parameters such as Net Present Value associated with a reserve entity?

RESENT VALUE.NET PRESENT VALUE tracks the value of a reserve entity based on a specific ECONOMIC SCENARIO.

How much more money would I have to spend to increase my return?

RESENT VALUE.CAPITAL COST tracks the future capital investment in a reserve entity based on a specific ECONOMIC SCENARIO.

What events in the field are related to the reserves information?

The revision category associated with the reserve revision volumes identifies the generic business process responsible for changes in the reserve volumes. For specific remarks, use the remarks column.

How do I handle the supporting documents associate with a reserve entity?

The RM module will handle this. Please refer to the Records Management Module reference guide for more details at http://www.ppdm.org/ftp/member/v36.

How do I manage the obligations associated with a reserves volume?

The obligation module includes a foreign key for linking an obligation back to a specific reserves entity. Please refer to the Obligations Module reference guide for details at http://www.ppdm.org/ftp/member/v36.

Appendix A: Sample Queries

These sample queries have been developed using a subset of the requirements defined in the Business Requirements Document. Note that there are many ways to address the questions posed here, but we have tried to provide useful examples that illustrate the use of the data model. The PPDM Association does not provide any guarantee that these queries will satisfy your business requirements; they are for illustration only.

- **Versioning over time:** Many aspects of the oil and gas business have a strong time component. Users require information about how a business object was configured in the past, what it looks like now, and what it is expected to look like in the future (i.e., if a project is not active now, when was it in the past). If your queries need to address the situation as it is now, use the ACTIVE_IND you will find in many versioned tables. Using this flag helps ensure that you do not return data that is out of date.

- **Effective and Expiry dates:** Reserves volumes are expected to change over time. To manage reserve reporting cycles, the reserves data is usually updated only within the current reserves period. At the end of this period the reserve volumes are considered to be frozen. Subsequent changes to these volumes then occur in a subsequent reserves period. To manage the concept of a reserves reporting period, most reserves tables also use a effective date and expiry date. To ensure that a query returns data for a specific period of time, add selection criteria to the effect that the date of interest is greater or equal to the effective date of the data, and less than or equal to the expiry date of the data.

- **Units of Measure:** Several examples have been provided to show how units of measure should be queried in PPDM. As these queries are nearly always handled the same way, this guide does not show the method every time it is needed; the authors felt that this would create confusion and obscure the main intent of the query.

Standard oil and gas reports.

Reporting is the cornerstone to managing corporate reserves. To this end, two standard reserve entity reports have been created in Crystal Report one for an oil entity and one for a gas entity. The actual report data for the 2002 reserves year specifically is included in the appendices.

The reports are made up of data returned at an entity level, and data returned by sub reports, where more than one record is expected to be returned for a given entity. Where data is derived from a sub report, the data itself is enclosed in a grey box, and the sub report name is also indicated in Grey.

Gas Detail Report;

What is the detailed gas reserves data for this entity?

```
SELECT
    RS.RESENT_ID, RS.EFFECTIVE_DATE, RS.EXPIRY_DATE, RS.RESENT_NAME,
    RS.DESCRIPTION, RSC.RESERVE_CLASS_ID, RSC.EFFECTIVE_DATE, RSC.EXPIRY_DATE,
    RSP.EFFECTIVE_DATE, RSP.EXPIRY_DATE, RSSS.PRODUCT_TYPE,
    RSSS.CURRENT_BALANCE, RSSS.CURRENT_BALANCE_UOM, RSSS.EFFECTIVE_DATE,
    RSSS.EXPIRY_DATE, RSSS.GROSS_SUMMARY_IND, RSSS.OPEN_BALANCE,
    RSSS.OPEN_BALANCE_UOM, RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID,
    RSVS.PRODUCT_TYPE, RSVS.ANALYST_BA_ID, RSVS.APPROVED_BY_BA_ID,
    RSVS.APPROVED_DATE, RSVS.CURRENT_BALANCE, RSVS.CURRENT_BALANCE_UOM,
    RSVS.DECLINE_CASE_ID, RSVS.EFFECTIVE_DATE, RSVS.EXPIRY_DATE,
    RSVS.GROSS_SUMMARY_IND, RSVS.OPEN_BALANCE, RSVS.OPEN_BALANCE_UOM,
    RSVS.PDEN_ID, RSVS.PDEN_TYPE, RSVS.PDEN_PRODUCT_TYPE,
    RSVS.REMARK, RSVS.SOURCE, RSVS.VOLUME_METHOD,
    RPP.EFFECTIVE_DATE, RPP.EXPIRY_DATE, RPP.HEAT_CONTENT,
    RPP.HEAT_CONTENT_OUOM, RPP.LOSS_FACTOR, PDENVA.CASE_ID,
    PDENVA.AREA_SIZE, PDENVA.AREA_SIZE_OUOM, PDENVA.CASE_NAME,
    PDENVA.GAS_ABANDON_COMPRESS, PDENVA.GAS_ABANDON_PRESS,
    PDENVA.GAS_ABANDON_PRESS_OUOM, PDENVA.GAS_INIT_COMPRESS,
    PDENVA.GAS_INIT_PRESSURE, PDENVA.GAS_ORIGINAL_IN_PLACE,
    PDENVA.GAS_ORIGINAL_IN_PLACE_OUOM, PDENVA.GAS_RATIO_BGI, PDENVA.GAS_RECOVERY,
    PDENVA.GROSS_PAY, PDENVA.GROSS_PAY_OUOM, PDENVA.INIT_RES_TEMP,
    PDENVA.INIT_RES_TEMP_OUOM, PDENVA.NET_PAY, PDENVA.NET_PAY_OUOM,
    PDENVA.POROSITY, PDENVA.REMARK, PDENVA.WATER_SATURATION,
    PDENMB.CASE_ID, PDENMB.CASE_NAME, PDENMB.CO2_PERCENT,
    PDENMB.CRITICAL_PRESS, PDENMB.CRITICAL_PRESS_OUOM, PDENMB.CRITICAL_TEMP,
    PDENMB.CRITICAL_TEMP_OUOM, PDENMB.CUM_VOLUME, PDENMB.CUM_VOLUME_DATE,
    PDENMB.CUM_VOLUME_UOM, PDENMB.GAS_ABANDON_PRESS, PDENMB.GAS_ABANDON_PRESS_OUOM,
    PDENMB.GAS_ABANDON_RECOVER, PDENMB.H2S_PERCENT, PDENMB.INITIAL_CUM_VOLUME_OUOM,
    PDENMB.INITIAL_CUM_VOLUME_UOM, PDENMB.INITIAL_PRESS, PDENMB.INITIAL_PRESS_OUOM,
    PDENMB.INITIAL_TEMP, PDENMB.INITIAL_TEMP_OUOM, PDENMB.N2_PERCENT,
    PDENMB.ORIG_GAS_IN_PLACE, PDENMB.POOL_DATUM_DEPTH, PDENMB.POOL_DATUM_DEPTH_OUOM,
    PDENMB.RECOV_GAS_IN_PLACE, PDENMB.SPECIFIC_GRAVITY
FROM
    RESERVE_ENTITY RS,RESENT_CLASS RSC,RESENT_PRODUCT RSP,RESENT_VOL_SUMMARY RSSS,
    RESENT_VOL_SUMMARY RSVS,RESENT_PROD_PROPERTY RPP,PDEN_VOLUME_ANALYSIS PDENVA,
    PDEN_MATERIAL_BAL PDENMB
WHERE
    RS.RESENT_ID = RSC.RESENT_ID AND
    RSC.RESENT_ID = RSP.RESENT_ID AND
    RSC.RESERVE_CLASS_ID = RSP.RESERVE_CLASS_ID AND
    RSC.RESENT_ID = RSSS.RESENT_ID(+) AND
    RSC.RESERVE_CLASS_ID = RSSS.RESERVE_CLASS_ID(+) AND
    RSP.RESENT_ID = RSVS.RESENT_ID AND
    RSP.RESERVE_CLASS_ID = RSVS.RESERVE_CLASS_ID AND
    RSP.PRODUCT_TYPE = RSVS.PRODUCT_TYPE AND
    RSP.RESENT_ID = RPP.RESENT_ID AND
    RSP.RESERVE_CLASS_ID = RPP.RESERVE_CLASS_ID AND
    RSP.PRODUCT_TYPE = RPP.PRODUCT_TYPE AND
    RSVS.PDEN_ID = PDENVA.PDEN_ID(+) AND
    RSVS.PDEN_TYPE = PDENVA.PDEN_TYPE(+) AND
    RSVS.PDEN_SOURCE = PDENVA.PDEN_SOURCE(+) AND
    RSVS.PDEN_PRODUCT_TYPE = PDENVA.PRODUCT_TYPE(+) AND
    RSVS.VOL_ANAL_CASE_ID = PDENVA.CASE_ID(+) AND
    RSVS.PDEN_ID = PDENMB.PDEN_ID(+) AND
    RSVS.PDEN_TYPE = PDENMB.PDEN_TYPE(+) AND
    RSVS.PDEN_SOURCE = PDENMB.PDEN_SOURCE(+) AND
    RSVS.PDEN_PRODUCT_TYPE = PDENMB.PRODUCT_TYPE(+) AND
    RSVS.MATERIAL_BALANCE_CASE_ID = PDENMB.CASE_ID(+) AND
    RSVS.PRODUCT_TYPE = 'GAS' AND
    RSVS.GROSS_SUMMARY_IND = 'Y' AND
    RS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
```

```
    RSC.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSP.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RPP.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSC.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSP.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RPP.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSSS.PRODUCT_TYPE = 'SALES GAS' AND
    RSSS.GROSS_SUMMARY_IND = 'Y' AND
    RSSS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSSS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
ORDER BY
    RS.RESENT_ID ASC, RSC.RESERVE_CLASS_ID ASC
```

Oil Detail Report;

What is the detailed oil reserves data for this entity?

```
SELECT
    RS.RESENT_ID, RS.EFFECTIVE_DATE, RS.EXPIRY_DATE, RS.RESENT_NAME,
    RS.DESCRIPTION, RSC.RESERVE_CLASS_ID, RSC.EFFECTIVE_DATE, RSC.EXPIRY_DATE,
    RSP.EFFECTIVE_DATE, RSP.EXPIRY_DATE, RSVS.RESENT_ID,
    RSVS.RESERVE_CLASS_ID, RSVS.PRODUCT_TYPE, RSVS.ANALYST_BA_ID,
    RSVS.APPROVED_BY_BA_ID, RSVS.APPROVED_DATE, RSVS.CURRENT_BALANCE,
    RSVS.CURRENT_BALANCE_UOM, RSVS.DECLINE_CASE_ID, RSVS.EFFECTIVE_DATE,
    RSVS.EXPIRY_DATE, RSVS.GROSS_SUMMARY_IND, RSVS.OPEN_BALANCE,
    RSVS.OPEN_BALANCE_UOM, RSVS.PDEN_ID, RSVS.PDEN_TYPE,
    RSVS.PDEN_PRODUCT_TYPE, RSVS.REMARK, RSVS.SOURCE,
    RSVS.VOLUME_METHOD, RPP.EFFECTIVE_DATE, RPP.EXPIRY_DATE,
    RPP.OIL_DENSITY, RPP.OIL_DENSITY_OUOM, RPP.SULPHUR_CONTENT,
    RPP.SULPHUR_CONTENT_OUOM, PDENVA.AREA_SIZE, PDENVA.AREA_SIZE_OUOM,
    PDENVA.CASE_NAME, PDENVA.GAS_ABANDON_PRESS_OUOM, PDENVA.GAS_INIT_PRESSURE,
    PDENVA.GROSS_PAY, PDENVA.GROSS_PAY_OUOM, PDENVA.INIT_RES_TEMP,
    PDENVA.INIT_RES_TEMP_OUOM, PDENVA.NET_PAY, PDENVA.NET_PAY_OUOM,
    PDENVA.OIL_IN_PLACE, PDENVA.OIL_IN_PLACE_OUOM, PDENVA.OIL_ORIGINAL_IN_PLACE,
    PDENVA.OIL_ORIGINAL_IN_PLACE_OUOM, PDENVA.OIL_RECOVERY_PRIMARY,
    PDENVA.OIL_RECOVERY_SECONDARY, PDENVA.OIL_RECOVERY_TOTAL,
    PDENVA.OIL_RESIDUAL_SAT, PDENVA.OIL_SHRINKAGE, PDENVA.ORIGINAL_GOR,
    PDENVA.ORIG_SOL_GAS_IN_PLACE, PDENVA.ORIG_SOL_GAS_IN_PLACE_OUOM,
    PDENVA.POROSITY, PDENVA.PROJECT_ID, PDENVA.RECOV_GOR,
    PDENVA.RECOV_SOL_GAS_IN_PLACE, PDENVA.RECOV_SOL_GAS_IN_PLACE_OUOM,
    PDENVA.REMARK, PDENVA.SOL_GAS_RECOVERY, PDENVA.WATER_SATURATION,
    PDENVA.ORIGINAL_GOR_OUOM
FROM
    RESERVE_ENTITY RS,RESENT_CLASS RSC,RESENT_PRODUCT RSP,RESENT_VOL_SUMMARY RSVS,
    RESENT_PROD_PROPERTY RPP,PDEN_VOLUME_ANALYSIS PDENVA
WHERE
    RS.RESENT_ID = RSC.RESENT_ID AND
    RSC.RESENT_ID = RSP.RESENT_ID AND
    RSC.RESERVE_CLASS_ID = RSP.RESERVE_CLASS_ID AND
    RSP.RESENT_ID = RSVS.RESENT_ID AND
    RSP.RESERVE_CLASS_ID = RSVS.RESERVE_CLASS_ID AND
    RSP.PRODUCT_TYPE = RSVS.PRODUCT_TYPE AND
    RSP.RESENT_ID = RPP.RESENT_ID(+) AND
    RSP.RESERVE_CLASS_ID = RPP.RESERVE_CLASS_ID(+) AND
    RSP.PRODUCT_TYPE = RPP.PRODUCT_TYPE(+) AND
    RSVS.PDEN_ID = PDENVA.PDEN_ID(+) AND
    RSVS.PDEN_TYPE = PDENVA.PDEN_TYPE(+) AND
    RSVS.PDEN_SOURCE = PDENVA.PDEN_SOURCE(+) AND
    RSVS.PDEN_PRODUCT_TYPE = PDENVA.PRODUCT_TYPE(+) AND
    RSVS.VOL_ANAL_CASE_ID = PDENVA.CASE_ID(+) AND
    RSVS.PDEN_PRODUCT_TYPE = 'OIL' AND
```

```
    RSVS.GROSS_SUMMARY_IND = 'Y' AND
    RS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSC.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSP.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RPP.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSC.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSP.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RPP.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
   ORDER BY
    RS.RESENT_ID ASC, RSC.RESERVE_CLASS_ID ASC
```

Sub Report BYPRODUCTS: what byproducts are associated with this entity on a gross basis?

```
SELECT
    RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID,
    RSVS.PRODUCT_TYPE, RSVS.CURRENT_BALANCE,
    RSVS.CURRENT_BALANCE_UOM, RSVS.EFFECTIVE_DATE,
    RSVS.EXPIRY_DATE, RSVS.GROSS_SUMMARY_IND,
    RSVS.YIELD_PARENT_PRODUCT, RSVS.YIELD_RATE,
    RSVS.YIELD_UOM
FROM
    RESENT_VOL_SUMMARY RSVS
WHERE
    RSVS.RESENT_ID = '3128' AND
    RSVS.RESERVE_CLASS_ID = 'P+PDP' AND
    RSVS.GROSS_SUMMARY_IND = 'Y' AND
    RSVS.EFFECTIVE_DATE<=TO_DATE ('31-12-2001','DD-MM-YYYY') AND
    RSVS.EXPIRY_DATE>=TO_DATE ('31-12-2001','DD-MM-YYYY')
ORDER BY
    RSVS.PRODUCT_TYPE ASC
```

Sub Report Components: What components are associated with this reserve entity?

```
SELECT
    RSCOM.RESENT_ID, RSCOM.COMPONENT_ID, RSCOM.FIELD_ID, RSCOM.POOL_ID,
    RSCOM.UWI, FLD.ACTIVE_IND, FLD.FIELD_NAME, PL.ACTIVE_IND, PL.POOL_NAME,
    W.CURRENT_STATUS, W.WELL_NAME
FROM
    RESENT_COMPONENT RSCOM, FIELD FLD, POOL PL, WELL W
WHERE
    RSCOM.FIELD_ID = FLD.FIELD_ID(+) AND
    RSCOM.SOURCE = FLD.SOURCE(+) AND
    RSCOM.POOL_ID = PL.POOL_ID(+) AND
    RSCOM.SOURCE = PL.SOURCE(+) AND
    RSCOM.UWI = W.UWI(+) AND
    RSCOM.RESENT_ID = '3128'
```

Sub Report DECLINE: What are the decline forecast segments associated with this entity for the product GAS?

```
SELECT
    PDENDFC.PDEN_ID, PDENDFC.PDEN_TYPE, PDENDFC.PRODUCT_TYPE, PDENDFC.CASE_ID,
    PDENDFC.CASE_NAME, PDENDFC.REMARK, PDENDF.PDEN_SOURCE, PDENDF.CASE_ID,
    PDENDF.DECLINE_CURVE_TYPE, PDENDF.END_DATE, PDENDF.FINAL_DECLINE,
    PDENDF.FINAL_RATE, PDENDF.INITIAL_DECLINE, PDENDF.INITIAL_RATE,
```

```
    PDENDF.MINIMUM_DECLINE, PDENDF.N_FACTOR, PDENDF.RATE_OUOM,
    PDENDF.RATIO_FINAL_RATE, PDENDF.RATIO_FLUID_TYPE, PDENDF.RATIO_INITIAL_RATE,
    PDENDF.RATIO_RATE_OUOM, PDENDF.START_DATE, PDENDF.VOLUME,
    PDENDF.VOLUME_OUOM
FROM
    PDEN_DECLINE_CASE PDENDFC,PDEN_DECLINE_SEGMENT PDENDF
WHERE
    PDENDFC.PDEN_ID = PDENDF.PDEN_ID AND
    PDENDFC.PDEN_TYPE = PDENDF.PDEN_TYPE AND
    PDENDFC.PDEN_SOURCE = PDENDF.PDEN_SOURCE AND
    PDENDFC.PRODUCT_TYPE = PDENDF.PRODUCT_TYPE AND
    PDENDFC.CASE_ID = PDENDF.CASE_ID AND
    PDENDFC.PDEN_ID = '3128' AND
    PDENDFC.PDEN_TYPE = 'PDEN_RESENT' AND
    PDENDFC.PRODUCT_TYPE = 'GAS' AND
    PDENDF.PDEN_SOURCE = 'PPDM' AND
    PDENDF.CASE_ID = '5'
```

Sub Report Gross Revisions: What gross revision volumes have changed my summary volumes for this reporting period?

```
SELECT
    RSVR.RESENT_ID, RSVR.RESERVE_CLASS_ID, RSVR.PRODUCT_TYPE,
    RSVR.REVISION_OBS_NO, RSVR.APPROVED_BY_BA_ID, RSVR.EFFECTIVE_DATE,
    RSVR.EXPIRY_DATE, RSVR.GROSS_REVISION_IND, RSVR.NEW_VOLUME,
    RSVR.NEW_VOLUME_UOM, RSVR.REVISION_CATEGORY_ID, RSVR.REVISION_DATE,
    RSVR.REVISION_VOLUME, RSVR.REVISION_VOLUME_UOM, RSRC.CATEGORY_TYPE
FROM
    RESENT_VOL_REVISION RSVR, RESENT_REVISION_CAT RSRC
WHERE
    RSVR.REVISION_CATEGORY_ID = RSRC.REVISION_CATEGORY_ID AND
    RSVR.GROSS_REVISION_IND = RSRC.GROSS_IND AND
    RSVR.SOURCE = RSRC.SOURCE AND
    RSVR.RESENT_ID = '3128' AND
    RSVR.RESERVE_CLASS_ID = 'P+PDP' AND
    RSVR.EFFECTIVE_DATE >= TO_DATE ('01-01-2002', 'DD-MM-YYYY') AND
    RSVR.EFFECTIVE_DATE <= TO_DATE ('01-01-2002', 'DD-MM-YYYY') AND
    RSVR.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVR.EXPIRY_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVR.GROSS_REVISION_IND = 'Y'
ORDER BY
    RSVR.PRODUCT_TYPE ASC,RSVR.REVISION_OBS_NO ASC
```

Sub Report HIERARCHY: Where on the corporate reporting hierarchies does this entity fit?

```
SELECT
    RSRH.HIERARCHY_NAME, RSHD.LEVEL_SEQ_NO,
    RSHD.LEVEL_NAME, RSHL.AREA_ID,
    RSHL.COMPONENT_TYPE, RSHL.FIELD_ID,
    RSHL.POOL_ID, A.PREFERRED_NAME, PL.POOL_NAME,
    FLD.FIELD_NAME, RSHU.RESENT_ID
FROM
    RESENT_REPORT_HIER RSRH,RESERVE_HIER_DESC RSHD,RESENT_HIER_LEVEL RSHL,
    AREA A, POOL PL,FIELD FLD,RESENT_HIER_USE RSHU
WHERE
    RSRH.REPORT_HIERARCHY_ID = RSHD.RESERVE_HIERARCHY_ID AND
    RSHD.RESERVE_HIERARCHY_ID = RSHL.REPORT_HIERARCHY_ID AND
    RSHD.LEVEL_TYPE = RSHL.COMPONENT_TYPE AND
    RSHL.AREA_ID = A.AREA_ID(+) AND
    RSHL.POOL_ID = PL.POOL_ID(+) AND
    RSHL.FIELD_ID = FLD.FIELD_ID(+) AND
    RSHL.REPORT_HIERARCHY_ID = RSHU.REPORT_HIERARCHY_ID AND
```

```
    RSHL.COMPONENT_ID = RSHU.COMPONENT_ID AND
    RSHU.RESENT_ID = '3128'
ORDER BY
    RSRH.HIERARCHY_NAME ASC,
    RSHD.LEVEL_SEQ_NO ASC
```

Sub Report NET BYPRODUCTS: what are the byproducts associated with this entity on a net working interest basis?

```
SELECT
    RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID, RSVS.PRODUCT_TYPE, RSVS.CURRENT_BALANCE,
    RSVS.CURRENT_BALANCE_UOM, RSVS.EFFECTIVE_DATE, RSVS.EXPIRY_DATE,
RSVS.NET_SUMMARY_IND,
    RSVS.OPEN_BALANCE, RSVS.OPEN_BALANCE_UOM, RSVS.YIELD_PARENT_PRODUCT,
    BA.BUSINESS_ASSOCIATE, BA.BA_NAME
FROM
    RESENT_VOL_SUMMARY RSVS,BUSINESS_ASSOCIATE BA
WHERE
    RSVS.INTEREST_SET_PARTNER = BA.BUSINESS_ASSOCIATE AND
    RSVS.SOURCE = BA.SOURCE AND
    RSVS.RESENT_ID = '3128' AND
    RSVS.RESERVE_CLASS_ID = 'P+PDP' AND
    RSVS.NET_SUMMARY_IND = 'Y' AND
    RSVS.EFFECTIVE_DATE<=TO_DATE ('31-12-2001','DD-MM-YYYY') AND
    RSVS.EXPIRY_DATE>=TO_DATE ('31-12-2001','DD-MM-YYYY')
ORDER BY
    BA.BUSINESS_ASSOCIATE ASC, RSVS.PRODUCT_TYPE ASC
```

Sub Report NET GAS: What are the net gas ownership volumes associated with this entity?

```
SELECT
    RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID, RSVS.PRODUCT_TYPE, RSVS.CURRENT_BALANCE,
    RSVS.CURRENT_BALANCE_UOM, RSVS.EFFECTIVE_DATE, RSVS.EXPIRY_DATE,
RSVS.NET_SUMMARY_IND,
    RSVS.OPEN_BALANCE, RSVS.OPEN_BALANCE_UOM, BA.BA_NAME, RSSS.PRODUCT_TYPE,
    RSSS.CURRENT_BALANCE, RSSS.CURRENT_BALANCE_UOM, RSSS.EFFECTIVE_DATE,
RSSS.EXPIRY_DATE,
    RSSS.NET_SUMMARY_IND, RSSS.OPEN_BALANCE, RSSS.OPEN_BALANCE_UOM,
    INTP.GROSS_PERCENT_INTEREST, INT.INTEREST_SET_TYPE
FROM
    RESENT_VOL_SUMMARY RSVS, BUSINESS_ASSOCIATE BA, RESENT_VOL_SUMMARY RSSS,
    INT_SET_PARTNER INTP, INTEREST_SET INT
WHERE
    RSVS.INTEREST_SET_PARTNER = BA.BUSINESS_ASSOCIATE AND
    RSVS.SOURCE = BA.SOURCE AND
    RSVS.RESENT_ID = RSSS.RESENT_ID(+) AND
    RSVS.RESERVE_CLASS_ID = RSSS.RESERVE_CLASS_ID(+) AND
    RSVS.INTEREST_SET_ID = RSSS.INTEREST_SET_ID(+) AND
    RSVS.INTEREST_SET_PARTNER = RSSS.INTEREST_SET_PARTNER(+) AND
    RSVS.INTEREST_SET_SEQ_NO = RSSS.INTEREST_SET_SEQ_NO(+) AND
    RSVS.INTEREST_SET_ID = INTP.INTEREST_SET_ID AND
    RSVS.INTEREST_SET_SEQ_NO = INTP.INTEREST_SET_SEQ_NO AND
    RSVS.INTEREST_SET_PARTNER = INTP.PARTNER_BA_ID AND
    RSVS.SOURCE = INTP.SOURCE AND
    INTP.INTEREST_SET_ID = INT.INTEREST_SET_ID AND
    INTP.INTEREST_SET_SEQ_NO = INT.INTEREST_SET_SEQ_NO AND
    RSVS.RESENT_ID = '3128' AND
    RSVS.RESERVE_CLASS_ID = 'P+PDP' AND
    RSVS.PRODUCT_TYPE = 'GAS' AND
    RSVS.NET_SUMMARY_IND = 'Y' AND
    RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2001', 'DD-MM-YYYY') AND
    RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2001', 'DD-MM-YYYY') AND
```

```
    RSSS.PRODUCT_TYPE = 'SALES GAS' AND
    RSSS.NET_SUMMARY_IND = 'Y' AND
    RSSS.EFFECTIVE_DATE <= TO_DATE ('31-12-2001', 'DD-MM-YYYY') AND
    RSSS.EXPIRY_DATE >= TO_DATE ('31-12-2001', 'DD-MM-YYYY')
ORDER BY
    BA.BA_NAME ASC
```

Sub Report NET OIL: What are the net oil ownership volumes associated with this entity?

```
SELECT
    RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID, RSVS.PRODUCT_TYPE, RSVS.CURRENT_BALANCE,
    RSVS.CURRENT_BALANCE_UOM, RSVS.EFFECTIVE_DATE, RSVS.EXPIRY_DATE,
RSVS.NET_SUMMARY_IND,
    RSVS.OPEN_BALANCE, RSVS.OPEN_BALANCE_UOM, BA.BA_NAME, INTP.GROSS_PERCENT_INTEREST,
    INT.INTEREST_SET_TYPE
FROM
    RESENT_VOL_SUMMARY RSVS, BUSINESS_ASSOCIATE BA,
    INT_SET_PARTNER INTP, INTEREST_SET INT
WHERE
    RSVS.INTEREST_SET_PARTNER = BA.BUSINESS_ASSOCIATE AND
    RSVS.SOURCE = BA.SOURCE AND
    RSVS.INTEREST_SET_ID = INTP.INTEREST_SET_ID AND
    RSVS.INTEREST_SET_SEQ_NO = INTP.INTEREST_SET_SEQ_NO AND
    RSVS.INTEREST_SET_PARTNER = INTP.PARTNER_BA_ID AND
    RSVS.SOURCE = INTP.SOURCE AND
    INTP.INTEREST_SET_ID = INT.INTEREST_SET_ID AND
    INTP.INTEREST_SET_SEQ_NO = INT.INTEREST_SET_SEQ_NO AND
    RSVS.RESENT_ID = '132' AND
    RSVS.RESERVE_CLASS_ID = 'PDP' AND
    RSVS.PRODUCT_TYPE = 'LIGHT OIL' AND
    RSVS.NET_SUMMARY_IND = 'Y' AND
    RSVS.EFFECTIVE_DATE<=TO_DATE ('31-12-2002','DD-MM-YYYY') AND
    RSVS.EXPIRY_DATE>=TO_DATE ('31-12-2002','DD-MM-YYYY')
ORDER BY
    BA.BA_NAME ASC
```

Sub Report NET REVISIONS: What are the net revision volumes associated with the entity for the companies with an interest position?.

```
SELECT
    RSVR.RESENT_ID, RSVR.RESERVE_CLASS_ID, RSVR.PRODUCT_TYPE,
    RSVR.REVISION_OBS_NO, RSVR.APPROVED_BY_BA_ID, RSVR.EFFECTIVE_DATE,
    RSVR.EXPIRY_DATE, RSVR.NET_REVISION_IND, RSVR.NEW_VOLUME,
    RSVR.NEW_VOLUME_UOM, RSVR.REVISION_CATEGORY_ID, RSVR.REVISION_DATE,
    RSVR.REVISION_VOLUME, RSVR.REVISION_VOLUME_UOM, RSRC.CATEGORY_TYPE, BA.BA_NAME
FROM
    RESENT_VOL_REVISION RSVR,RESENT_REVISION_CAT RSRC,BUSINESS_ASSOCIATE BA
WHERE
    RSVR.REVISION_CATEGORY_ID = RSRC.REVISION_CATEGORY_ID AND
    RSVR.ACTIVE_IND = RSRC.ACTIVE_IND AND
    RSVR.NET_REVISION_IND = RSRC.NET_IND AND
    RSVR.SOURCE = RSRC.SOURCE AND
    RSVR.PARTNER_BA_ID = BA.BUSINESS_ASSOCIATE AND
    RSVR.SOURCE = BA.SOURCE AND
    RSVR.RESENT_ID = '3128' AND
    RSVR.RESERVE_CLASS_ID = 'P+PDP' AND
    RSVR.EFFECTIVE_DATE >= TO_DATE ('01-01-2002', 'DD-MM-YYYY') AND
    RSVR.EFFECTIVE_DATE <= TO_DATE ('01-01-2002', 'DD-MM-YYYY') AND
    RSVR.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVR.EXPIRY_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY') AND
    RSVR.NET_REVISION_IND = 'Y'
ORDER BY
    BA.BA_NAME ASC,RSVR.PRODUCT_TYPE ASC,RSVR.REVISION_OBS_NO ASC
```

How do my economic reserve volumes compare to my technical reserve volumes?

```
SELECT
    RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID, RSVS.PRODUCT_TYPE,
    RSVS.CURRENT_BALANCE TECHNICAL_RESERVES,
    RSVS.CURRENT_BALANCE_UOM TECHNICAL_RESERVES_UOM,
    RSV.RESERVE_LIFE_INDEX ECONOMIC_RESERVE_LIFE,RSV.TECH_FORECAST
TECHNICAL_RESERVE_LIFE,
    RSES.REMAINING_BALANCE ECONOMIC_RESERVES,
    RSES.REMAINING_BALANCE_UOM ECONOMIC_RESERVES_UOM
FROM
    RESENT_VOL_SUMMARY RSVS, RESENT_ECO_VOLUME RSES, RESENT_ECO_RUN RSV
WHERE
    RSVS.EFFECTIVE_DATE <= TO_DATE('31-DEC-2002','DD-MON-YYYY')
    AND RSVS.EXPIRY_DATE >= TO_DATE('31-DEC-2002','DD-MON-YYYY')
    AND RSVS.GROSS_SUMMARY_IND = 'Y'
    AND RSV.EFFECTIVE_DATE <= TO_DATE('31-DEC-2002','DD-MON-YYYY')
    AND RSV.EXPIRY_DATE >= TO_DATE('31-DEC-2002','DD-MON-YYYY')
    AND RSES.EFFECTIVE_DATE <= TO_DATE('31-DEC-2002','DD-MON-YYYY')
    AND RSES.EXPIRY_DATE >= TO_DATE('31-DEC-2002','DD-MON-YYYY')
    AND RSVS.RESENT_ID = RSV.RESENT_ID
    AND RSVS.RESERVE_CLASS_ID = RSV.RESERVE_CLASS_ID
    AND RSES.ECONOMICS_RUN_ID = RSV.ECONOMICS_RUN_ID
    AND RSVS.RESENT_ID = RSES.RESENT_ID
    AND RSVS.RESERVE_CLASS_ID = RSES.RESERVE_CLASS_ID
     AND RSVS.PRODUCT_TYPE = RSES.PRODUCT_TYPE
```

How do I use the PRODUCT_COMP table to aggregate reserves products up to my reporting products ?

```
SELECT
   RSVS.RESERVE_CLASS_ID, PRODC.PRODUCT_TYPE ,
   ROUND(SUM(RSVS.OPEN_BALANCE * TO_NUMBER(PRODC.FORMULA)),3) OPENING_BALANCE,
   ROUND(SUM(RSVS.CURRENT_BALANCE * TO_NUMBER(PRODC.FORMULA)),3) CURRENT_BALANCE,
   MIN(RSVS.CURRENT_BALANCE_UOM) UOM
FROM
    RESENT_VOL_SUMMARY RSVS, PRODUCT_COMP PRODC
WHERE
   RSVS.PRODUCT_TYPE = PRODC.SUBPRODUCT_TYPE
   AND RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
   AND RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
   AND RSVS.GROSS_SUMMARY_IND = 'Y'
   AND RSVS.RESERVE_CLASS_ID = 'PDP'
GROUP BY
   RSVS.RESERVE_CLASS_ID, PRODC.PRODUCT_TYPE
```

How do I use the Reserve Class Formula and Calculation tables to report my reserves volumes at a specific reserve class (in this case "P+1/2P" ?

```
SELECT
    RESENT_ID, AAARCF.RESERVE_CLASS_ID RESERVE_CLASS,
    PRODUCT_TYPE,  SUM(CURRENT_BALANCE * CONTRIBUTION_FACTOR) CURRENT_BALLANCE,
    MAX(CURRENT_BALANCE_UOM) CURRENT_UOM,
    SUM(OPEN_BALANCE * CONTRIBUTION_FACTOR) OPEN_BALANCE, MAX(OPEN_BALANCE_UOM) OPEN_UOM,
    MAX(PDEN_PRODUCT_TYPE) PDEN_PRODUCT_TYPE,
    MAX(YIELD_PARENT_PRODUCT) YIELD_PARENT_PRODUCT
FROM
    RESENT_VOL_SUMMARY RSVS, RESERVE_CLASS_CALC AAARCC, RESERVE_CLASS_FORMULA AAARCF
WHERE
    AAARCF.RESERVE_CLASS_ID = 'P+1/2P'
    AND AAARCC.RESERVE_CLASS_ID = AAARCF.RESERVE_CLASS_ID
    AND AAARCC.FORMULA_ID = AAARCF.FORMULA_ID
```

```
    AND AAARCC.ORIGIN_RESERVE_CLASS_ID = RSVS.RESERVE_CLASS_ID
    AND PPDM_UTILITIES.CHECK_RULE(RSVS.RESENT_ID, TO_DATE ('31-12-2002', 'DD-MM-YYYY'),
    AAARCF.USE_RULE) = 'YES'
    AND RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
    AND RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
    AND RSVS.GROSS_SUMMARY_IND = 'Y'
GROUP BY
    RESENT_ID, AAARCF.RESERVE_CLASS_ID, PRODUCT_TYPE
```

How do I allocate reserve volumes back to the components making up my reserve entity which have a defined PERCENT_CONTRIBUTION?

```
SELECT
    RSVS.RESENT_ID, RSVS.RESERVE_CLASS_ID, RSVS.PRODUCT_TYPE,
    RSCOM.UWI, RSCOM.FIELD_ID, RSCOM.POOL_ID,
    ROUND(RSCOM.PERCENT_CONTRIBUTION,4) PERCENT_CONTRIBUTION,
    ROUND(RSVS.CURRENT_BALANCE*RSCOM.PERCENT_CONTRIBUTION/100,3) ALLOCATED_RESERVES,
    RSVS.CURRENT_BALANCE_UOM
FROM
    RESENT_VOL_SUMMARY RSVS, RESENT_COMPONENT RSCOM
WHERE
    RSVS.RESENT_ID = RSCOM.RESENT_ID
    AND RSCOM.ACTIVE_IND = 'Y'
    AND RSCOM.PERCENT_CONTRIBUTION > 0
    AND RSVS.RESENT_ID = '3128'
    AND RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
    AND RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
    AND RSVS.PRODUCT_TYPE = 'SALES GAS'
    AND RSVS.RESERVE_CLASS_ID = 'PDP'
    AND RSVS.GROSS_SUMMARY_IND = 'Y'
```

How do I select the current production volumes for all the wells making up my reserve entity, as well as any forecasted volumes for those months lacking production data.

```
SELECT
    P.RESENT_ID, P.RESERVE_CLASS_ID RESERVE_CLASS, P.PRODUCT_TYPE,
    MAX(P.CYR_PROD) CYR_PROD, MAX(P.PROD_UNITS) PROD_UNITS,
    MAX(P.PROD_END_DATE) PROD_END_DATE, SUM(VOLUME) FORECAST_VOLUME,
    MAX(VOLUME_OUOM) FORECAST_UNITS
FROM
    PDEN_DECLINE_SEGMENT PDENDF,
    (SELECT RESENT_ID, R.PDEN_ID, PDENVBM.PRODUCT_TYPE, MAX(VOLUME_END_DATE)
    PROD_END_DATE, SUM(YTD_VOLUME) CYR_PROD, MAX(VOLUME_UOM) PROD_UNITS,
RESERVE_CLASS_ID,
    DECLINE_CASE_ID
    FROM PDEN_VOL_BY_MONTH PDENVBM,
    (SELECT DISTINCT RS.RESENT_ID, RSVS.PDEN_ID, UWI, RSVS.PDEN_PRODUCT_TYPE,
    RESERVE_CLASS_ID, DECLINE_CASE_ID FROM RESERVE_ENTITY RS, RESENT_COMPONENT RSCOM,
    RESENT_VOL_SUMMARY RSVS
    WHERE RS.RESENT_ID = RSCOM.RESENT_ID
        AND RS.RESENT_ID = RSVS.RESENT_ID
        AND RS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
        AND RS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
        AND RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
        AND RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
        AND GROSS_SUMMARY_IND = 'Y'
        AND UWI IS NOT NULL) R
WHERE
    PDENVBM.PRODUCT_TYPE = R.PDEN_PRODUCT_TYPE
    AND PDENVBM.PDEN_ID = R.UWI
    AND YEAR = TO_NUMBER('2002')
GROUP BY
    RESENT_ID, R.PDEN_ID, RESERVE_CLASS_ID, DECLINE_CASE_ID, PDENVBM.PRODUCT_TYPE) P
WHERE
```

```
    P.PDEN_ID=PDENDF.PDEN_ID(+)
    AND P.PRODUCT_TYPE=PDENDF.PRODUCT_TYPE(+)
    AND P.PROD_END_DATE < START_DATE(+)
    AND P.DECLINE_CASE_ID = CASE_ID(+)
    AND TO_DATE ('31-12-2002', 'DD-MM-YYYY') >= END_DATE(+)
GROUP BY
    P.RESENT_ID, P.PRODUCT_TYPE, P.RESERVE_CLASS_ID
```

OK now how do I use the volume conversion tables to ensure that my historic production volumes and my forecast production volumes are in the same unit of measure?

```
SELECT
    P.RESENT_ID, P.RESERVE_CLASS_ID RESERVE_CLASS, P.PRODUCT_TYPE,
    PPDM_UTILITIES.CONVERSION(P.PRODUCT_TYPE,MAX(P.CYR_PROD),MAX(P.PROD_UNITS),
    decode(P.PRODUCT_TYPE, 'GAS', 'MMcf','OIL','MBbl','E3m3'),
    P.RESENT_ID,to_date('31-DEC-2002','DD-MON-YYYY'))  CYR_PROD,
    decode(P.PRODUCT_TYPE, 'GAS', 'MMcf','OIL','MBbl','E3m3') PROD_UNITS,
    MAX(P.PROD_END_DATE) PROD_END_DATE,
    PPDM_UTILITIES.CONVERSION(P.PRODUCT_TYPE, SUM(VOLUME), MAX(VOLUME_OUOM),
    decode(P.PRODUCT_TYPE, 'GAS', 'MMcf','OIL','MBbl','E3m3'),
    P.RESENT_ID,to_date('31-DEC-2002','DD-MON-YYYY')) FORECAST_VOLUME,
    decode(P.PRODUCT_TYPE, 'GAS', 'MMcf','OIL','MBbl','E3m3') FORECAST_UNITS
FROM
    PDEN_DECLINE_SEGMENT PDENDF, (SELECT RESENT_ID, R.PDEN_ID, PDENVBM.PRODUCT_TYPE,
    MAX(VOLUME_END_DATE) PROD_END_DATE, SUM(YTD_VOLUME) CYR_PROD,
    MAX(VOLUME_UOM) PROD_UNITS, RESERVE_CLASS_ID, DECLINE_CASE_ID
    FROM
       PDEN_VOL_BY_MONTH PDENVBM, (SELECT DISTINCT RS.RESENT_ID, RSVS.PDEN_ID,
       UWI, RSVS.PDEN_PRODUCT_TYPE, RESERVE_CLASS_ID, DECLINE_CASE_ID
       FROM
           RESERVE_ENTITY RS, RESENT_COMPONENT RSCOM, RESENT_VOL_SUMMARY RSVS
       WHERE RS.RESENT_ID = RSCOM.RESENT_ID
           AND RS.RESENT_ID = RSVS.RESENT_ID
           AND RS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
           AND RS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
           AND RSVS.EFFECTIVE_DATE <= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
           AND RSVS.EXPIRY_DATE >= TO_DATE ('31-12-2002', 'DD-MM-YYYY')
           AND GROSS_SUMMARY_IND = 'Y'
           AND UWI IS NOT NULL) R
   WHERE
       PDENVBM.PRODUCT_TYPE = R.PDEN_PRODUCT_TYPE AND PDENVBM.PDEN_ID = R.UWI
       AND YEAR = TO_NUMBER('2002')
   GROUP BY
       RESENT_ID, R.PDEN_ID, RESERVE_CLASS_ID, DECLINE_CASE_ID, PDENVBM.PRODUCT_TYPE) P
WHERE
    P.PDEN_ID=PDENDF.PDEN_ID(+)
    AND P.PRODUCT_TYPE=PDENDF.PRODUCT_TYPE(+)
    AND P.PROD_END_DATE < START_DATE(+)
    AND P.DECLINE_CASE_ID = CASE_ID(+)
    AND TO_DATE ('31-12-2002', 'DD-MM-YYYY') >= END_DATE(+)
GROUP BY
    P.RESENT_ID, P.PRODUCT_TYPE, P.RESERVE_CLASS_ID
```

How do I convert units where I have multiple types of units present?
Note that the Resent_ID and Reporting_date may both be omitted from the Conversion Function if they are not required for the conversion (last two arguments). Even the unit code such as `'AREA'` or `'PRESSURE'` (first argument) is optional.

```
SELECT
    PDEN_ID, PDEN_TYPE, PDEN_SOURCE, PRODUCT_TYPE, CASE_ID,
    PPDM_UTILITIES.CONVERSION('AREA', AREA_SIZE, AREA_SIZE_OUOM, 'Acres' ) AREA_SIZE,
    'Acres' AREA_UOM, CASE_NAME,
```

```
    PPDM_UTILITIES.CONVERSION('PRESSURE', GAS_ABANDON_PRESS, GAS_ABANDON_PRESS_OUOM,
    'Psi') GAS_ABANDON_PRESS,
    'Psi' PRESS_OUOM,
    PPDM_UTILITIES.CONVERSION('TEMP',INIT_RES_TEMP, INIT_RES_TEMP_OUOM, 'F')
     RESERVOIR_TEMP, 'F' TEMP_UOM,
    PPDM_UTILITIES.CONVERSION(PRODUCT_TYPE, GAS_ORIGINAL_IN_PLACE,
    GAS_ORIGINAL_IN_PLACE_OUOM, 'Bcf', PDEN_ID, null) GAS_ORIGINAL_IN_PLACE,
   'Bcf' GAS_IN_PLACE_UOM,
    PPDM_UTILITIES.CONVERSION('LENGTH', NET_PAY, NET_PAY_OUOM, 'Ft') NET_PAY, 'Ft'
     PAY_UOM,
    PPDM_UTILITIES.CONVERSION(PRODUCT_TYPE, OIL_IN_PLACE, OIL_IN_PLACE_OUOM, 'MBbl',
     PDEN_ID, to_date('31-DEC-2002','DD-MON-YYYY')) OIL_IN_PLACE,'MBbl' OIL_IN_PLACE_UOM,
    PPDM_UTILITIES.CONVERSION('GOR', ORIGINAL_GOR, ORIGINAL_GOR_OUOM, 'CF/BBL')
     ORIGINAL_GOR,'SCF/Bbl' GOR_UOM
FROM PDEN_VOLUME_ANALYSIS
```

How do I report the economic summary results on an 8/8ths (gross lease) basis for an economic run? Note that data with a forward slash in the units of measure such as oil price ($/Bbl) is averaged, while the rest of the data is summed. Only records where the period_type is "YEAR" or null is selected to avoid double counting monthly and yearly data for the same time period.

```
SELECT
    PARTNER_BA_ID, RESENT_ID, RESERVE_CLASS_ID, EXPIRY_DATE,
    PRODUCT_TYPE, ECONOMIC_SCHEDULE, SCHEDULE_VALUE_UOM,
    DECODE(INSTR(SCHEDULE_VALUE_UOM, '/'), 0, SUM_VALUE, ROUND(AVG_VALUE,3)) VALUE,
    SCHEDULE_DESC
FROM
   (SELECT 'GROSS LEASE' PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID,
        RSV.EXPIRY_DATE, PRODUCT_TYPE, ECONOMIC_SCHEDULE,
        MAX(SCHEDULE_VALUE_UOM) SCHEDULE_VALUE_UOM,
        SUM(SCHEDULE_VALUE) SUM_VALUE, AVG(SCHEDULE_VALUE) AVG_VALUE,
        MAX(SCHEDULE_DESC) SCHEDULE_DESC
    FROM
        RESENT_ECO_RUN RSV, RESENT_ECO_SCHEDULE RSESC
    WHERE
        RSV.RESENT_ID = RSESC.RESENT_ID
        AND RSV.RESERVE_CLASS_ID = RSESC.RESERVE_CLASS_ID
        AND RSV.ECONOMICS_RUN_ID = RSESC.ECONOMICS_RUN_ID
        AND GROSS_IND = 'Y'
        AND NVL(PERIOD_TYPE, 'YEAR') = 'YEAR'
        AND INSTR(ECONOMIC_SCHEDULE, 'CUM') = 0
GROUP BY
    RSV.RESENT_ID, RSV.RESERVE_CLASS_ID,
    RSV.EXPIRY_DATE, PRODUCT_TYPE, ECONOMIC_SCHEDULE)
```

How do I report the economic summary results for the Business Associates for an economic run?

```
SELECT
    PARTNER_BA_ID, RESENT_ID, RESERVE_CLASS_ID, EXPIRY_DATE,
    PRODUCT_TYPE, ECONOMIC_SCHEDULE, SCHEDULE_VALUE_UOM,
    DECODE(INSTR(SCHEDULE_VALUE_UOM, '/'), 0, SUM_VALUE, ROUND(AVG_VALUE,3)) VALUE,
    SCHEDULE_DESC
FROM
   (SELECT
        PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID,
        RSV.EXPIRY_DATE, PRODUCT_TYPE, ECONOMIC_SCHEDULE,
        MAX(SCHEDULE_VALUE_UOM) SCHEDULE_VALUE_UOM,
```

```
        SUM(SCHEDULE_VALUE) SUM_VALUE, AVG(SCHEDULE_VALUE) AVG_VALUE,
        MAX(SCHEDULE_DESC) SCHEDULE_DESC
    FROM
        RESENT_ECO_RUN RSV, RESENT_ECO_SCHEDULE RSESC
    WHERE
        RSV.RESENT_ID = RSESC.RESENT_ID
        AND RSV.RESERVE_CLASS_ID = RSESC.RESERVE_CLASS_ID
        AND RSV.ECONOMICS_RUN_ID = RSESC.ECONOMICS_RUN_ID
        AND NET_IND = 'Y'
        AND NVL(PERIOD_TYPE, 'YEAR') = 'YEAR'
        AND INSTR(ECONOMIC_SCHEDULE, 'CUM') = 0
    GROUP BY
        RSV.PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID,
        RSV.EXPIRY_DATE, PRODUCT_TYPE, ECONOMIC_SCHEDULE)
```

How do I report the yearly economic results on an 8/8ths (gross lease) basis for an economic run? This query could also be used to set up a view in order to act as a Yearly table.

```
SELECT
    'GROSS LEASE' PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID,
    RSV.EXPIRY_DATE, PRODUCT_TYPE, ECONOMIC_SCHEDULE, SCHEDULE_DATE,
    SCHEDULE_VALUE_UOM, SCHEDULE_VALUE
FROM
    RESENT_ECO_RUN RSV, RESENT_ECO_SCHEDULE RSESC
WHERE
    RSV.RESENT_ID = RSESC.RESENT_ID
    AND RSV.RESERVE_CLASS_ID = RSESC.RESERVE_CLASS_ID
    AND RSV.ECONOMICS_RUN_ID = RSESC.ECONOMICS_RUN_ID
    AND GROSS_IND = 'Y'
    AND PERIOD_TYPE = 'YEAR'
    AND INSTR(ECONOMIC_SCHEDULE, 'CUM') = 0
ORDER BY
    RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, PRODUCT_TYPE, ECONOMIC_SCHEDULE, SCHEDULE_DATE
```

How do I report the yearly economic results for the Business Associates for an economic run? ? This query could also be used to set up a view in order to act as a Yearly table.

```
SELECT
    PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, RSV.EXPIRY_DATE, PRODUCT_TYPE,
    ECONOMIC_SCHEDULE, SCHEDULE_DATE, SCHEDULE_VALUE_UOM, SCHEDULE_VALUE
FROM
    RESENT_ECO_RUN RSV, RESENT_ECO_SCHEDULE RSESC
WHERE
    RSV.RESENT_ID = RSESC.RESENT_ID
    AND RSV.RESERVE_CLASS_ID = RSESC.RESERVE_CLASS_ID
    AND RSV.ECONOMICS_RUN_ID = RSESC.ECONOMICS_RUN_ID
    AND NET_IND = 'Y'
    AND PERIOD_TYPE = 'YEAR'
    AND INSTR(ECONOMIC_SCHEDULE, 'CUM') = 0
ORDER BY
    RSV.PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, PRODUCT_TYPE,
    ECONOMIC_SCHEDULE, SCHEDULE_DATE
```

How do I report the monthly economic results on an 8/8ths (gross lease) basis for an economic run? This query could also be used to set up a view in order to act as a monthly budget table.

```
SELECT
```

```
    'GROSS LEASE' PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, RSV.EXPIRY_DATE,
    M.PRODUCT_TYPE, M.ECONOMIC_SCHEDULE, M.SCHED_DATE,M.SCHEDULE_VALUE_UOM,
    M1.SCHEDULE_VALUE JAN, M2.SCHEDULE_VALUE FEB, M3.SCHEDULE_VALUE MAR,
    M4.SCHEDULE_VALUE APR, M5.SCHEDULE_VALUE MAY, M6.SCHEDULE_VALUE JUN,
    M7.SCHEDULE_VALUE JUL, M8.SCHEDULE_VALUE AUG, M9.SCHEDULE_VALUE SEP,
    M10.SCHEDULE_VALUE OCT, M11.SCHEDULE_VALUE NOV, M12.SCHEDULE_VALUE DEC
FROM
   RESENT_ECO_RUN RSV, (
   SELECT DISTINCT RESENT_ID, RESERVE_CLASS_ID, ECONOMICS_RUN_ID, EXPIRY_DATE,
       PRODUCT_TYPE, ECONOMIC_SCHEDULE, SCHEDULE_VALUE_UOM,
       SUBSTR(SCHEDULE_DATE,1,4) SCHED_DATE
   FROM
       RESENT_ECO_SCHEDULE RSESC
   WHERE
       PERIOD_TYPE = 'MONTH') M,
   RESENT_ECO_SCHEDULE M1,  RESENT_ECO_SCHEDULE M2,  RESENT_ECO_SCHEDULE M3,
   RESENT_ECO_SCHEDULE M4,  RESENT_ECO_SCHEDULE M5,  RESENT_ECO_SCHEDULE M6,
   RESENT_ECO_SCHEDULE M7,  RESENT_ECO_SCHEDULE M8,  RESENT_ECO_SCHEDULE M9,
   RESENT_ECO_SCHEDULE M10,  RESENT_ECO_SCHEDULE M11,  RESENT_ECO_SCHEDULE M12
WHERE
    GROSS_IND = 'Y'
    AND RSV.RESENT_ID = M.RESENT_ID
    AND RSV.RESERVE_CLASS_ID = M.RESERVE_CLASS_ID
    AND RSV.ECONOMICS_RUN_ID = M.ECONOMICS_RUN_ID
    AND M.RESENT_ID = M1.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M1.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M1.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M2.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M2.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M2.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M3.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M3.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M3.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M4.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M4.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M4.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M5.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M5.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M5.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M6.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M6.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M6.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M7.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M7.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M7.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M8.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M8.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M8.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M9.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M9.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M9.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M10.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M10.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M10.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M11.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M11.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M11.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M12.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M12.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M12.ECONOMICS_RUN_ID(+)
    AND M.SCHED_DATE || '/1' = M1.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/2' = M2.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/3' = M3.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/4' = M4.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/5' = M5.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/6' = M6.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/7' = M7.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/8' = M8.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/9' = M9.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/10' = M10.SCHEDULE_DATE(+)
```

```
    AND M.SCHED_DATE || '/11' = M11.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/12' = M12.SCHEDULE_DATE(+)
    AND M.PRODUCT_TYPE = M1.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M2.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M3.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M4.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M5.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M6.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M7.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M8.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M9.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M10.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M11.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M12.PRODUCT_TYPE(+)
    AND M.ECONOMIC_SCHEDULE = M1.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M2.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M3.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M4.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M5.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M6.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M7.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M8.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M9.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M10.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M11.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M12.ECONOMIC_SCHEDULE(+)
ORDER BY
    RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, M.PRODUCT_TYPE, M.ECONOMIC_SCHEDULE,
M.SCHED_DATE
```

How do I report the monthly economic results for the Business Associates for an economic run? ? This query could also be used to set up a view in order to act as a monthly budget table.

```
SELECT
    PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, RSV.EXPIRY_DATE, M.PRODUCT_TYPE,
    M.ECONOMIC_SCHEDULE, M.SCHED_DATE,M.SCHEDULE_VALUE_UOM, M1.SCHEDULE_VALUE JAN,
    M2.SCHEDULE_VALUE FEB, M3.SCHEDULE_VALUE MAR, M4.SCHEDULE_VALUE APR,
    M5.SCHEDULE_VALUE MAY, M6.SCHEDULE_VALUE JUN, M7.SCHEDULE_VALUE JUL,
    M8.SCHEDULE_VALUE AUG, M9.SCHEDULE_VALUE SEP, M10.SCHEDULE_VALUE OCT,
    M11.SCHEDULE_VALUE NOV, M12.SCHEDULE_VALUE DEC
FROM
    RESENT_ECO_RUN RSV, (
    SELECT DISTINCT RESENT_ID, RESERVE_CLASS_ID, ECONOMICS_RUN_ID, EXPIRY_DATE,
        PRODUCT_TYPE, ECONOMIC_SCHEDULE, SCHEDULE_VALUE_UOM,
        SUBSTR(SCHEDULE_DATE,1,4) SCHED_DATE
    FROM
        RESENT_ECO_SCHEDULE WHERE PERIOD_TYPE = 'MONTH') M,
    RESENT_ECO_SCHEDULE M1,  RESENT_ECO_SCHEDULE M2,  RESENT_ECO_SCHEDULE M3,
    RESENT_ECO_SCHEDULE M4,  RESENT_ECO_SCHEDULE M5,  RESENT_ECO_SCHEDULE M6,
    RESENT_ECO_SCHEDULE M7,  RESENT_ECO_SCHEDULE M8,  RESENT_ECO_SCHEDULE M9,
    RESENT_ECO_SCHEDULE M10,  RESENT_ECO_SCHEDULE M11,  RESENT_ECO_SCHEDULE M12
WHERE
    NET_IND = 'Y'
    AND RSV.RESENT_ID = M.RESENT_ID
    AND RSV.RESERVE_CLASS_ID = M.RESERVE_CLASS_ID
    AND RSV.ECONOMICS_RUN_ID = M.ECONOMICS_RUN_ID
    AND M.RESENT_ID = M1.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M1.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M1.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M2.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M2.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M2.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M3.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M3.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M3.ECONOMICS_RUN_ID(+)
```

```
    AND M.RESENT_ID = M4.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M4.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M4.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M5.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M5.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M5.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M6.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M6.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M6.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M7.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M7.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M7.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M8.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M8.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M8.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M9.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M9.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M9.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M10.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M10.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M10.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M11.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M11.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M11.ECONOMICS_RUN_ID(+)
    AND M.RESENT_ID = M12.RESENT_ID(+)
    AND M.RESERVE_CLASS_ID = M12.RESERVE_CLASS_ID(+)
    AND M.ECONOMICS_RUN_ID = M12.ECONOMICS_RUN_ID(+)
    AND M.SCHED_DATE || '/1' = M1.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/2' = M2.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/3' = M3.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/4' = M4.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/5' = M5.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/6' = M6.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/7' = M7.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/8' = M8.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/9' = M9.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/10' = M10.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/11' = M11.SCHEDULE_DATE(+)
    AND M.SCHED_DATE || '/12' = M12.SCHEDULE_DATE(+)
    AND M.PRODUCT_TYPE = M1.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M2.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M3.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M4.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M5.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M6.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M7.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M8.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M9.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M10.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M11.PRODUCT_TYPE(+)
    AND M.PRODUCT_TYPE = M12.PRODUCT_TYPE(+)
    AND M.ECONOMIC_SCHEDULE = M1.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M2.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M3.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M4.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M5.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M6.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M7.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M8.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M9.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M10.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M11.ECONOMIC_SCHEDULE(+)
    AND M.ECONOMIC_SCHEDULE = M12.ECONOMIC_SCHEDULE(+)
ORDER BY
    RSV.PARTNER_BA_ID, RSV.RESENT_ID, RSV.RESERVE_CLASS_ID, M.PRODUCT_TYPE,
    M.ECONOMIC_SCHEDULE, M.SCHED_DATE
```

This is our wish list of things to be stored in a probabilistic module in PPDM: a min(P90), most likely(P50) and maximum values(P10) for reservoir area, net pay, porosity, water saturation, hydrocarbon recovery factor and recoverable reserves.

Reserve classes may be defined for P10, P50 and P90 just as they may be defined for Proved, Probable and Possible. It will be up to each implementation to determine which classes are defined; the confidence type associated with each class; and the formulas to be used to determine the sum or difference between two or more classes. A development indicator may be used to differentiate between Proved Developed and Proved Undeveloped classes, or between P90 Developed and P90 Undeveloped if this is applicable. There is also a production indicator to subdivide Developed classes into Producing and Non Producing.

If required, an implementation may define both the traditional deterministic reserve classes, as well as probabilistic classes. Where a database holds a mix of deterministic and probabilistic classes, the confidence factor in the RESENT_CLASS table supports conversion between deterministic and probabilistic classes.

For each reserve volume summary record, which is specific to an entity, fluid and reserve class, there may be a corresponding entry in the PDEN VOLUME ANALYSIS table. This table holds columns for storing the volumetric data for reservoir area, net pay, porosity, water saturation, hydrocarbon recovery factor and recoverable reserves etc. In this fashion the min(P90), most likely(P50) and maximum values(P10) for the recoverable reserve volumes and associated volumetric data may be stored for each fluid associated with each reserve entity.

What code is in the utilities package that handles volume conversions and reserve class validation?

```
CREATE OR REPLACE PACKAGE PPDM_UTILITIES IS

function CHECK_RULE
  (p_RESENT_ID  IN VARCHAR2,
  p_EVALUATION_DATE IN DATE,
  p_STATEMENT VARCHAR2
  ) return varchar2;

function CONVERSION
  (p_PRODUCT_TYPE IN VARCHAR2,
  p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2,
  p_RESENT_ID in VARCHAR2,
  p_EVALUATION_DATE in DATE
  ) return number;

function CONVERSION
  (p_PRODUCT_TYPE IN VARCHAR2,
  p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2
  ) return number;

function CONVERSION
```

```
  (p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2,
  p_RESENT_ID in VARCHAR2,
  p_EVALUATION_DATE in DATE
  ) return number;

function CONVERSION
  (p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2
  ) return number;

function CONVERSION
  (p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2
  ) return number;

END PPDM_UTILITIES;

/
CREATE OR REPLACE PACKAGE BODY PPDM_UTILITIES AS

function CONVERSION
  (p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2,
  p_RESENT_ID in VARCHAR2,
  p_EVALUATION_DATE in DATE
  ) return number is

  begin

       return(ppdm_utilities.conversion(null, p_value, p_UOM_IN, p_UOM_OUT, p_RESENT_ID,
p_EVALUATION_DATE));

  end;

  function CONVERSION
  (p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2
  ) return number is

  begin

       return(ppdm_utilities.conversion(null, p_value, p_UOM_IN, p_UOM_OUT, null, null));

  end;


  function CONVERSION
  (p_PRODUCT_TYPE IN VARCHAR2,
  p_value IN NUMBER,
  p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2
  ) return number is

  begin

       return(ppdm_utilities.conversion(p_PRODUCT_TYPE, p_value, p_UOM_IN, p_UOM_OUT,
null, null));

  end;

  function CONVERSION
  ( p_UOM_IN  IN VARCHAR2,
  p_UOM_OUT IN VARCHAR2
  ) return number is
```

```
  begin

       return(ppdm_utilities.conversion(NULL, 1, p_UOM_IN, p_UOM_OUT, null, null));

  end;

  function CONVERSION
    (p_PRODUCT_TYPE IN VARCHAR2,
    p_value IN NUMBER,
    p_UOM_IN  IN VARCHAR2,
    p_UOM_OUT IN VARCHAR2,
    p_RESENT_ID IN VARCHAR2,
    p_EVALUATION_DATE IN DATE
    ) return number is

    line_count number(3);
    s_factor number(20,10);
    s_offset number(20,10);
    s_formula varchar2(200);
    resultant number(20,10);
    s_conv_quant varchar2(24);
    s_prod_value number(20,10) default 1;
    s_regime varchar2(24);
    inv_flag boolean default false;

    begin

      if upper(p_UOM_IN) = upper(p_UOM_OUT) then return(p_value); end if;

      select count(*) into line_count from PRODUCT
      where upper(PRODUCT_TYPE) = upper(p_PRODUCT_TYPE);

      if nvl(line_count,0) = 0 then -- This is not a recognized PRODUCT type, must be a
unit quantity

         select count(*) into line_count from PPDM_UNIT_CONVERSION
         where upper(UNIT_QUANTITY) = upper(p_PRODUCT_TYPE);

         if nvl(line_count,0) = 0 then -- This is not a recognized unit quantity, check
the to and from units anyways

              begin
                select nvl(CONVERSION_OFFSET,0), UNIT_EXPRESSION into s_offset, s_factor
from PPDM_UNIT_CONVERSION
                where upper(FROM_UNIT_SYMBOL) = upper(p_UOM_IN)
                and upper(TO_UNIT_SYMBOL) = upper(p_UOM_OUT)
                and rownum = 1;
              exception
              when others then -- assume not found and try going the other way
                 begin
                    select -nvl(CONVERSION_OFFSET,0)/UNIT_EXPRESSION, 1/UNIT_EXPRESSION
into s_offset, s_factor from PPDM_UNIT_CONVERSION
                    where upper(FROM_UNIT_SYMBOL) = upper(p_UOM_OUT)
                    and upper(TO_UNIT_SYMBOL) = upper(p_UOM_IN)
                    and rownum = 1;
                 exception
                 when others then

                    begin
                      select CONVERSION_FACTOR, 0 into s_factor, s_offset
                      from PPDM_VOL_MEAS_CONV
                      where upper(FROM_UOM) = upper(p_UOM_IN)
                      and upper(TO_UOM) = upper(p_UOM_OUT)
                      and rownum = 1;
                    exception
                    when others then -- assume not found and try going the other way
                       begin
                                select 1/CONVERSION_FACTOR, 0 into s_factor, s_offset
                                from PPDM_VOL_MEAS_CONV
                                where upper(FROM_UOM) = upper(p_UOM_OUT)
```

```
                              and upper(TO_UOM) = upper(p_UOM_IN)
                              and rownum = 1;
                        exception
                        when others then
                           return(null);  -- give up here
                        end;
                     end;

                  end;
               end;

            else -- This is a recognized unit quantity, check the to and from units

               begin
                 select nvl(CONVERSION_OFFSET,0), UNIT_EXPRESSION into s_offset, s_factor
from PPDM_UNIT_CONVERSION
                 where upper(FROM_UNIT_SYMBOL) = upper(p_UOM_IN)
                 and upper(TO_UNIT_SYMBOL) = upper(p_UOM_OUT)
                 and upper(UNIT_QUANTITY) = upper(p_PRODUCT_TYPE)
                 and rownum = 1;
               exception
               when others then -- assume not found and try going the other way
                 begin
                   select -nvl(CONVERSION_OFFSET,0)/UNIT_EXPRESSION, 1/UNIT_EXPRESSION into
s_offset, s_factor from PPDM_UNIT_CONVERSION
                   where upper(FROM_UNIT_SYMBOL) = upper(p_UOM_OUT)
                   and upper(TO_UNIT_SYMBOL) = upper(p_UOM_IN)
                   and upper(UNIT_QUANTITY) = upper(p_PRODUCT_TYPE)
                   and rownum = 1;
                 exception
                 when others then
                   return(null);  -- give up here
                 end;
               end;

            end if;

            resultant := p_value * s_factor + s_offset;
            return (ROUND(resultant,4));

      else -- This is a Product Type, so now get the CONVERSION_QUANTITY

             select CONVERSION_QUANTITY into s_conv_quant from PRODUCT
             where upper(PRODUCT_TYPE) = upper(p_PRODUCT_TYPE)
             and rownum = 1;

             begin
               select VOLUME_REGIME_ID into s_regime from RESENT_VOL_REGIME
               where RESENT_ID = p_resent_id
               and ACTIVE_IND = 'Y'
               and rownum = 1;
             exception
             when others then
                 begin
                   select VOLUME_REGIME_ID into s_regime from PDEN_VOL_REGIME
                   where PDEN_ID = p_resent_id
                   and ACTIVE_IND = 'Y'
                   and rownum = 1;
                 exception
                 when others then
                    s_regime := null;
                 end;
             end;

             if s_regime is null then

                 begin
                   select CONVERSION_FACTOR, upper(CONVERSION_FORMULA) into s_factor,
s_formula
                   from PPDM_VOL_MEAS_CONV
```

```
                  where upper(FROM_UOM) = upper(p_UOM_IN)
                  and upper(TO_UOM) = upper(p_UOM_OUT)
                  and upper(CONVERSION_QUANTITY) = upper(s_conv_quant)
                  and rownum = 1;
                  inv_flag := false;
               exception
               when others then -- assume not found and try going the other way
                  begin
                     select CONVERSION_FACTOR, upper(CONVERSION_FORMULA) into s_factor,
s_formula
                     from PPDM_VOL_MEAS_CONV
                     where upper(FROM_UOM) = upper(p_UOM_OUT)
                     and upper(TO_UOM) = upper(p_UOM_IN)
                     and upper(CONVERSION_QUANTITY) = upper(s_conv_quant)
                     and rownum = 1;
                     inv_flag := true;
                  exception
                  when others then
                     return(null);  -- give up here
                  end;
               end;

            else

               begin
                     select CONVERSION_FACTOR, upper(CONVERSION_FORMULA) into s_factor,
s_formula
                     from PPDM_VOL_MEAS_CONV
                     where upper(FROM_UOM) = upper(p_UOM_IN)
                     and upper(TO_UOM) = upper(p_UOM_OUT)
                     and upper(CONVERSION_QUANTITY) = upper(s_conv_quant)
                     and VOLUME_REGIME_ID = s_regime
                     and rownum = 1;
                     inv_flag := false;
               exception
               when others then -- assume not found and try going the other way
                     begin
                              select CONVERSION_FACTOR, upper(CONVERSION_FORMULA) into
s_factor, s_formula
                              from PPDM_VOL_MEAS_CONV
                              where upper(FROM_UOM) = upper(p_UOM_OUT)
                              and upper(TO_UOM) = upper(p_UOM_IN)
                              and upper(CONVERSION_QUANTITY) = upper(s_conv_quant)
                              and VOLUME_REGIME_ID = s_regime
                              and rownum = 1;
                              inv_flag := true;
                     exception
                     when others then
                        return(null);  -- give up here
                     end;
               end;

            end if;

            if p_EVALUATION_DATE is null then

               if instr(s_formula, 'RPP.HEAT_CONTENT') > 0 then
                  select HEAT_CONTENT into s_prod_value from RESENT_PROD_PROPERTY
                  where resent_id = p_resent_id
                  and rownum = 1;
               elsif instr(s_formula, 'RPP.OIL_DENSITY') > 0 then
                  select OIL_DENSITY into s_prod_value from RESENT_PROD_PROPERTY
                  where resent_id = p_resent_id
                  and rownum = 1;
               else
                  s_prod_value := 1;

               end if;

            else
```

```
                if instr(s_formula, 'RPP.HEAT_CONTENT') > 0 then
                    select HEAT_CONTENT into s_prod_value from RESENT_PROD_PROPERTY
                    where resent_id = p_resent_id
                    and EFFECTIVE_DATE <= p_EVALUATION_DATE
                    AND EXPIRY_DATE >= p_EVALUATION_DATE
                    and rownum = 1;
                elsif instr(s_formula, 'RPP.OIL_DENSITY') > 0 then
                    select OIL_DENSITY into s_prod_value from RESENT_PROD_PROPERTY
                    where resent_id = p_resent_id
                    and EFFECTIVE_DATE <= p_EVALUATION_DATE
                    AND EXPIRY_DATE >= p_EVALUATION_DATE
                    and rownum = 1;
                else
                    s_prod_value := 1;

                end if;

            end if;

            -- This function should probably check that the storage units for the
RESENT_PROD_PROPERTY
            -- and the units expected in the conversion formula match up, but I'm going
to assume
            -- that the data storage UOM are consistant between these two tables.

            if inv_flag then
               resultant := p_value / s_factor / s_prod_value;
            else
                resultant := p_value * s_factor * s_prod_value;
            end if;

            return (ROUND(resultant,4));


       end if;

       return(null); -- shouldn't ever get here!

    exception -- unanticipated error must have happened
    when others then
       return(null);
    end CONVERSION;


   function CHECK_RULE
     (p_RESENT_ID  IN VARCHAR2,
     p_EVALUATION_DATE IN DATE,
     p_STATEMENT VARCHAR2
     ) return varchar2 IS

     Class varchar2(12);
     Class_Count number(3);
     First_Quote number(3);
     Second_Quote number(3);
     Qualifier varchar2(40);
     Evaluation varchar2(40);
     sql_err varchar2(2000);
     Colon number(3);
     Statement_ varchar2(400);
     Condition varchar2(40);

     BEGIN

        Statement_ := upper(trim(p_Statement));

        If upper(nvl(Statement_,'YES')) = 'YES' or upper(Statement_) = 'USE' then
           Return('YES');
        else
           Evaluation := 'NO';
```

```
        end if;


        While Length(Statement_) > 0 loop

        Colon := instr(Statement_, ':');

        if Colon > 1 then
            Condition := trim(substr(Statement_, 1, Colon - 1));
            Statement_ := trim(substr(Statement_, Colon + 1, 400));

        else
            Condition := Statement_;
            Statement_ := '';
        end if;

        First_Quote := instr(Condition,'"',1,1);
        Second_Quote := instr(Condition,'"',1,2);
        Class := upper(substr(Condition, First_Quote + 1, Second_Quote - First_Quote -
1));
        Qualifier := upper(trim(substr(Condition, Second_Quote + 1,40)));

        Select Count(RESERVE_CLASS_ID) into Class_Count From RESENT_CLASS
        where RESENT_ID = p_RESENT_ID
        and RESERVE_CLASS_ID = Class
        and ACTIVE_IND = 'Y'
        and EFFECTIVE_DATE <= p_EVALUATION_DATE
        and EXPIRY_DATE >= p_EVALUATION_DATE;

        Class_Count := nvl(Class_Count,0);

        if (substr(Qualifier, 1, 6) = 'EXISTS' or substr(Qualifier, 1, 11) = 'DOES
EXISTS') and Class_Count > 0 then
           Evaluation := 'YES';
        elsif (INSTR(Qualifier, 'MISSING') > 0 or INSTR(Qualifier, 'NOT') > 0) and
Class_Count = 0 then
           Evaluation := 'YES';
        else
           Return('NO');
        end if;

        End Loop;

        RETURN(Evaluation);

     exception
     when others then
         sql_err := 'Error: ' || sqlerrm;
         Return(sql_err);

    END CHECK_RULE;

END PPDM_UTILITIES;
/
```