

Система очередей заданий QMAN

QMAN - сокращение от Queue Manager.

1. Общее описание

Назначение системы – поддержка запуска и параллельного выполнения взаимозависимых заданий, организованных в очереди.

Очереди являются самостоятельными, независимыми друг от друга единицами, каждая со своим набором заданий, своими настройками и расписанием запуска. У каждого задания есть ссылка на PL/SQL-процедуру, которая выполняется во время запуска задания.

Все задания очереди организованы в группы. Каждое задание принадлежит какой-либо группе, заданий без групп не существует.

Группы заданий в составе очереди выполняются последовательно друг за другом, согласно заданного порядка. До тех пор, пока все задания в группе успешно не завершатся, выполнение следующей группы не начинается.

Задания в группах могут выполняться как последовательно друг за другом (синхронно), так и параллельно (асинхронно). При этом существует лимит на количество одновременно выполняющихся параллельных заданий.

Для параллельных заданий могут быть установлены взаимные зависимости, если требуется, чтобы перед запуском определенных заданий уже были успешно завершены их родительские задания.

Для предотвращения автоматического запуска очередей и заданий существует возможность их отключения. Если отключение очереди происходит во время ее выполнения, то это приводит к ее приостановке.

Существует возможность отметить задание как пропускаемое. При запуске такого задания на выполнение, связанная с ним PL/SQL-процедура не вызывается и задание сразу же завершается с успешным статусом. Возможен также одиночный пропуск задания, который полезен при восстановлении заданий после сбоев.

2. Программный интерфейс

Управление системой осуществляется через вызов процедур пакета QMAN_PKG. Весь программный интерфейс делится на три секции – управление метаданными, управление выполнением и поддержка мониторинга. Управление метаданными в свою очередь делится на наборы процедур, соответствующие очередям, группам заданий и заданиям.

Каждой очереди, группе и заданию соответствует уникальный символьный код, который выступает идентификатором объекта при вызове процедур интерфейса. Пространство имен для кодов групп и заданий ограничено их родительской очередью. Таким образом, в разных очередях могут быть группы с одинаковыми кодами, то же относится и к заданиям. Поэтому для идентификации этих видов объектов наряду с их кодом требуется также указывать код очереди.

Многие процедуры интерфейса управления требуют, чтобы очередь находилась в состоянии ожидания и при этом была отключена. Например, изменять геометрию очередей, т.е. создавать и удалять группы и задания, а также задавать приоритетность и зависимости между заданиями – допустимо только в промежутке между циклами выполнения очереди. Плюс к этому очередь должна быть отключена, чтобы исключить риск ее автоматического запуска в середине неатомарной операции, которую администратор может выполнять в этот момент путем последовательного вызова некоего набора интерфейсных процедур.

Для более подробного обзора интерфейсных процедур см. спецификацию пакета QMAN_PKG, где в комментариях кратко указано назначение каждой процедуры и ее параметров. Более подробное описание, сгруппированное по функциональному назначению, см. в соответствующих разделах настоящей документации.

3. Создание и удаление объектов системы

3.1. Создание очередей, заданий и их групп

Процедуры создания очередей, групп заданий и заданий имеют префикс CREATE_%:

Процедура	Описание
QMAN_PKG.CREATE_QUEUE	Создать очередь
QMAN_PKG.CREATE_GROUP	Создать группу заданий
QMAN_PKG.CREATE_TASK	Создать задание

В качестве аргументов требуется как минимум указать код нового объекта. А также, если это подразумевается типом объекта – код родителя, в пределах которого создается текущий объект (для групп и заданий) и обязательное описательное имя (для очередей и групп).

При создании задания обязательным параметром является имя PL/SQL-процедуры, которая будет выполняться при запуске задания. Имя должно быть задано однозначно, с указанием владельца схемы. Точка с запятой в конце не требуется. Система производит парсинг и проверку на наличие указанной процедуры в базе, и при каких-либо несоответствиях выдает ошибку.

Для исполняемой процедуры задания допускается задавать аргументы. Однако при этом никакой проверки на их корректность не производится, что содержит в себе потенциальный риск останова очереди при первом запуске нового задания. Следовательно, нужно быть достаточно внимательным при указании аргументов исполняемой PL/SQL-процедуры задания.

Новые группы при создании ставятся в конец очереди. Новые задания ставятся в конец группы. Если требуется иной порядок, то его нужно изменить дополнительным вызовом соответствующих интерфейсных процедур, описанных ниже.

По умолчанию, сразу после создания, все объекты отключены. Это обусловлено тем, что в общем случае полная настройка нового объекта после его создания выполняется при помощи дополнительного вызова различных процедур программного интерфейса. В этот промежуточный момент объект остается еще незавершенным и вводить его в строй нельзя. После полной настройки объекта, либо совокупности объектов, им нужно явно установить признак **enabled** при помощи вызова соответствующих интерфейсных процедур.

3.2. Изменение базовых характеристик объектов

Группе интерфейсных процедур, ответственных за создание объектов, соответствуют процедуры для изменения их базовых характеристик – уникального кода, описательного имени, а также исполняемой PL/SQL-процедуры заданий:

Процедура	Описание
QMAN_PKG.SET_QUEUE_CODE	Установить код очереди
QMAN_PKG.SET_QUEUE_NAME	Установить имя очереди
QMAN_PKG.SET_GROUP_CODE	Установить код группы заданий
QMAN_PKG.SET_GROUP_NAME	Установить имя группы заданий
QMAN_PKG.SET_TASK_CODE	Установить код задания
QMAN_PKG.SET_TASK_PROC	Установить исполняемую процедуру задания

Перепривязка объектов к новому родителю (перемещение групп между очередями, перемещение заданий между группами) не поддерживается ни в каком виде. Для переноса потребуется создать новый объект на новом месте и удалить старый.

3.3. Удаление объектов

Процедуры создания очередей, групп заданий и заданий имеют префикс DROP_%:

Процедура	Описание
QMAN_PKG.DROP_QUEUE	Удалить очередь
QMAN_PKG.DROP_GROUP	Удалить группу заданий
QMAN_PKG.DROP_TASK	Удалить задание

При удалении требуется соблюсти определенные условия: для очередей и групп необходимо отсутствие входящих в них групп и заданий, для заданий – отсутствие зависимых от них дочерних заданий.

4. Синхронный и асинхронный режимы выполнения

Задания могут выполняться либо в синхронном, либо в асинхронном режиме.

Синхронный режим подразумевает выполнение заданий последовательно друг за другом, согласно заданного порядка. До тех пор, пока задание не завершится успешно, следующие за ним не запустятся.

Асинхронный режим предназначен для параллельного выполнения заданий, в пределах заданного лимита одновременно выполняющихся заданий, согласно заданной приоритетности, и согласно заданным зависимостям заданий друг от друга.

Режим выполнения задается для группы. В очереди может быть несколько групп, каждая со своей настройкой режима. Группы всегда выполняются строго друг за другом, без взаимных пересечений, поэтому конфликтов разных режимов при работе очереди не происходит.

Режим задается путем установки либо сброса признака асинхронности **async** для группы заданий. При этом, если асинхронный режим для какой-либо группы требуется установить, то он должен быть также разрешен в целом для очереди, путем установки такого же признака. Таким образом, задания в группе начинают работать в асинхронном режиме только в том случае, если признак асинхронности **async** установлен как для самой группы заданий, так и в целом для очереди.

Процедуры установки или сброс признака асинхронности следующие:

Процедура	Описание
QMAN_PKG.SET_QUEUE_ASYNC	Установить признак асинхронности очереди
QMAN_PKG.SET_GROUP_ASYNC	Установить признак асинхронности группы заданий

4.1. Признак асинхронности заданий

Признак асинхронности также существует непосредственно для задания. Он имеет приоритет над признаками группы и очереди: когда он установлен, задание запускается в асинхронном режиме вне зависимости от настроек группы и очереди.

Если подобное задание входит в состав синхронной группы, то его поведение отличается от прочих заданий группы, у которых данный признак не установлен. Запускается оно согласно заданного порядка заданий в группе, однако очередь после этого не ожидает окончания его выполнения, для того чтобы запустить следующие за ним задания. Задание как бы «отпочковывается» от основной очереди выполнения, не влияя на запуск и выполнение прочих заданий группы. Тем не менее, если подобных заданий в группе несколько, все они подчиняются общему ограничению на количество одновременно выполняющихся заданий.

Если подобное задание входит в состав асинхронной группы, то ничего по своей сути при этом не меняется, т.к. группа и так является асинхронной.

Признак асинхронности для задания устанавливается так же, как и для очереди либо группы заданий:

Процедура	Описание
QMAN_PKG.SET_TASK_ASYNC	Установить признак асинхронности задания

5. Взаимные зависимости заданий

Взаимные зависимости заданий предназначены для асинхронного режима выполнения. Они обозначают, что такое-то задание (называемое дочерним) должно выполниться только после того, как успешно выполнится некий набор прочих заданий (называемых родительскими).

Для заданий в синхронном режиме зависимости ни в каком виде заданы быть не могут. Наличие таких зависимостей может привести к ступору очереди при синхронном выполнении заданий, когда более раннее по порядку задание зависит от более позднего. При этом ситуация штатными средствами исправлена быть не может, т.к. любое изменение геометрии очереди возможно только при выключенной очереди, находящейся при этом в режиме ожидания. В связи с этим, при попытке добавления зависимостей между заданиями синхронной группы, будет сгенерировано исключение. Также неуспешными будут попытки перевода группы либо очереди в синхронный режим из асинхронного, при наличии взаимных зависимостей между заданиями. Для этого потребуются сначала удалить все существующие зависимости.

Задание может иметь любое количество родительских заданий, либо не иметь вообще. У задания может быть любое количество дочерних заданий, либо не быть вообще. Очевидное условие – задание не может быть родителем самого себя, включая промежуточные поколения. Во время добавления взаимных связей через программный интерфейс производится проверка на то, что при этом не образуется циклическая зависимость и соответственно выдача ошибки при ее обнаружении.

Задание может быть связано зависимостями только с заданиями своей же группы. Зависимостей между заданиями разных групп и тем более очередей не может быть в принципе.

Добавление и удаление зависимостей осуществляется при помощи следующих процедур:

Процедура	Описание
QMAN_PKG.ADD_TASK_PARENT	Добавить зависимость задания
QMAN_PKG.REMOVE_TASK_PARENT	Удалить зависимость задания

6. Порядок и приоритетность выполнения заданий и групп

Понятие порядка выполнения применимо как для заданий, так и для их групп.

Применительно к заданиям порядок определяет, в какой последовательности они будут выполняться в синхронном режиме. Для асинхронного режима порядок выполнения приобретает смысл приоритетности. Чем меньше «порядковый номер» задания, тем раньше оно запустится на выполнение. Однако из-за наличия взаимных зависимостей заданий, в асинхронном режиме работы вполне возможны ситуации, когда задания, стоящие ближе к началу, могут запускаться позже тех, что стоят по порядку после них.

Применительно к группам заданий семантика та же, что и для синхронно выполняющихся заданий.

Изменение порядка выполнения при помощи интерфейсных процедур осуществляется путем указания объекта, после которого нужно поставить перемещаемый объект. Напрямую «порядковый номер» не задается. Если требуется поставить объект в самое начало последовательности, вместо предшествующего ему объекта указывается null.

Для заданий перемещения допустимы только в пределах своей группы.

Для групп перемещения допустимы только в том случае, если они не содержат в себе заданий. Это сделано для того, чтобы непреднамеренно не внести опасные изменения в уже настроенную очередь.

Процедуры изменения порядка выполнения следующие:

Процедура	Описание
QMAN_PKG.SET_GROUP_ORDER	Установить порядок выполнения группы заданий
QMAN_PKG.SET_TASK_ORDER	Установить порядок выполнения задания

6.1. Настройка приоритетности выполнения заданий в асинхронных группах

От того, как рассортированы задания в пределах асинхронной группы, может зависеть общее время ее выполнения. Если приоритетность заданий настроена неоптимально, то в какой-либо момент выполнения очереди может быть не полностью задействован ограниченный пул одновременно работающих заданий. Соответственно общее время выполнения группы при этом увеличивается.

Такое может случиться, например, если в группе существует несколько достаточно продолжительных заданий, приоритеты которых достаточно низки. Следствием этого будет слишком поздний запуск таких заданий, соответственно в конце выполнения группы останется всего одно-два долго выполняющихся задания, в то время как все прочие задания уже выполнены. В таком случае целесообразно передвинуть эти задания ближе к началу, для того чтобы они выполнялись параллельно с более мелкими и лимит одновременно выполняющихся заданий всегда был задействован полностью.

Подобная же ситуация может возникнуть с заданиями, связанными между собой зависимостями и которые при этом разбросаны по разным концам группы. В условиях ограниченного лимита на количество одновременно работающих заданий при этом возможен нежелательный простой и растягивание общего времени выполнения группы. Решением в данном случае будет компоновка взаимосвязанных заданий ближе друг к другу.

Настройка приоритетности с учетом всех условий может оказаться достаточно непростой задачей. Основным материалом для анализа здесь может послужить журнал выполнения за некоторый период, плюс информация об общей совокупности зависимостей заданий.

7. Лимит одновременно выполняющихся заданий

Лимит одновременно выполняющихся заданий предназначен в основном для групп заданий, настроенных на асинхронный режим. Однако он имеет влияние и на синхронные группы, содержащие в своем составе задания с установленным признаком асинхронности.

Лимит может быть задан как для каждой из групп, так и в целом для очереди. При запуске заданий берется минимальное значение из настроек ныне выполняющейся группы и очереди. Если какое-либо из этих двух значений не задано, берется то, которое задано. Если не задано ни одно, лимит принимается равным пяти.

Процедуры установки лимита одновременно выполняющихся заданий следующие:

Процедура	Описание
QMAN_PKG.SET_QUEUE_LIMIT	Установить порог одновременно работающих заданий очереди
QMAN_PKG.SET_TASK_LIMIT	Установить порог одновременно работающих заданий группы

8. Расписание выполнения очередей

Расписание выполнения очередей задается в виде списка часов по времени сервера, когда очередь должна автоматически запускаться. Часы перечисляются через запятую, диапазон допустимых значений от 0 до 23 включительно.

Для каждой очереди существует внутренняя переменная с типом `date`, которая задает время следующего ее запуска. При наступлении этого времени очередь запускается, при условии что она не отключена. Независимо от того, отключена она или нет, происходит обновление времени ее следующего запуска.

Если выполнение очереди затянулось настолько, что по наступлении времени следующего запуска она еще не завершена – это время не обновляется и остается прежним. Поэтому по окончании затянувшегося цикла очереди, следующий за ним запускается практически сразу же. Интервал проверки условий запуска очередей составляет 5 минут, новый цикл запустится в пределах этого времени.

Расписание выполнения очереди задается следующей процедурой:

Процедура	Описание
QMAN_PKG.SET_QUEUE_HOURS	Установить часы запуска очереди

8.1. Процедура проверки условий запуска очереди

Более сложные условия запуска можно моделировать при помощи пользовательской функции, которая должна возвращать значение с типом `boolean`. Функция задается в настройках очереди и запускается каждый раз при проверке условий запуска очереди, раз в 5 минут. В случае возврата `true` очередь будет запущена, в случае возврата `false` соответственно нет. В случае возврата `null` запуск будет осуществляться на основании алгоритма проверки расписания, описанного выше.

Независимо от того, задана пользовательская функция проверки условий запуска или нет, и независимо от того, что она возвращает – обновление времени следующего запуска очереди происходит всегда. Это позволяет гибко комбинировать оба способа проверки – «отключая» в нужное время пользовательскую процедуру, начав возвращать `null` значения из нее, и включая ее вновь в любое время.

Пользовательская функция проверки условий запуска очереди задается следующей процедурой:

Процедура	Описание
QMAN_PKG.SET_QUEUE_CHECK_PROC	Установить процедуру проверки условий запуска очереди

9. Управление выполнением очередей и заданий

9.1. Запуск очередей

Если возникла необходимость запуска очереди вне расписания, это можно сделать косвенно, установив дату следующего ее запуска при помощи следующей процедуры:

Процедура	Описание
QMAN_PKG.SET_QUEUE_NEXT	Установить дату/время следующего запуска очереди

Если установленная дата меньше или равна текущей, очередь запускается сразу же. В противном случае установленная дата будет проверяться и обрабатываться в штатном режиме алгоритмом запуска очереди по расписанию, который вызывается раз в 5 минут.

Всегда следует иметь в виду, что установленная дата влияет только на алгоритм запуска очереди по расписанию. Если для очереди определена пользовательская функция проверки условий запуска, она может оказать свое влияние таким образом, что запуска очереди не произойдет. В таком случае требуется либо убрать пользовательскую функцию из настроек очереди, либо изменить ее логику так, чтобы она в нужный момент вернула `null`.

Запуск очереди всегда выполняется асинхронно по отношению к текущей сессии. Все происходит аналогично тому, как если бы очередь была запущена автоматически. Мы только подгоняем все необходимые условия под то, чтобы алгоритмы проверки приняли решение в пользу запуска очереди в нужный нам момент. Поэтому даже при возникновении каких-либо сбоев при запуске, вызываемые интерфейсные процедуры не выдадут никаких ошибок или сообщений. Состояние только что запущенной очереди всегда следует дополнительно проконтролировать через предназначенные для этого средства мониторинга очередей и заданий.

9.2. Принудительный останов заданий

Выполняющееся задание может быть прервано при помощи следующей процедуры:

Процедура	Описание
QMAN_PKG.KILL_TASK	Прервать выполнение задания

При этом производится обрыв сессии задания путем `kill session immediate`, и в дальнейшем задание приобретает статус `BROKEN`.

9.3. Восстановление заданий после сбоев

Выполнение задания, которое было прервано по ошибке, может быть восстановлено путем вызова следующей процедуры:

Процедура	Описание
QMAN_PKG.RECOVER_TASK	Перезапустить задание после остановки (по ошибке)

При этом происходит перезапуск задания с самого начала. Очередь свое состояние восстанавливает автоматически.

Если специфика задания такова, что его нельзя перезапускать с самого начала, то администратор может вручную проделать все необходимые расчеты, начиная с прерванной точки до конца задания. После этого нужно будет установить признак одиночного пропуска для этого задания **skip** (см. ниже) и сделать восстановление его выполнения путем вызова процедуры, указанной выше. В таком случае задание в составе очереди выполнено не будет, очередь восстановит свое выполнение со следующих за ним заданий.

9.4. Пропуск выполнения заданий

Задание в очереди можно отметить как пропускаемое путем установки признака **bypass**. При запуске такого задания его исполняемая процедура не вызывается и оно сразу же отмечается как выполненное успешно. Данная возможность полезна для временного исключения заданий из очереди, либо для отключения заданий которые затем планируется удалить.

Процедура	Описание
QMAN_PKG.SET_TASK_BYPASS	Установить признак постоянного пропуска задания

Также существует возможность сделать одиночный пропуск задания путем установки признака **skip**. При запуске такого задания все происходит аналогично описанному выше, но при этом признак тут же сбрасывается, и при следующем запуске очереди задание пропущено уже не будет. Данная возможность полезна при восстановлении заданий после сбоев, когда все необходимые расчеты проведены вручную, задание в рамках очереди перезапускать не нужно и требуется только восстановить выполнение очереди с заданий, следующих за сбойным.

Процедура	Описание
QMAN_PKG.SET_TASK_SKIP	Установить признак одиночного пропуска задания

Технически признак одиночного пропуска не является специально выделенным признаком. Во время установки признака **skip** на самом деле присваивается специальное значение признаку **bypass**, равное 2. Что является знаком системе о том, что данное значение после обработки задания нужно сразу же сбросить.

9.5. Отключение очередей, заданий и их групп

Очереди, группы и задания могут быть отключены путем сброса признака **enabled**. Включать и отключать их можно в любое время, независимо от текущего состояния выполнения.

Процедуры установки значения признака **enabled** следующие:

Процедура	Описание
QMAN_PKG.SET_QUEUE_ENABLED	Установить признак включения очереди
QMAN_PKG.SET_GROUP_ENABLED	Установить признак включения группы заданий
QMAN_PKG.SET_TASK_ENABLED	Установить признак включения задания

Если отключена очередь, которая находится в состоянии ожидания, то она не может быть запущена ни автоматически, ни принудительно. Если очередь отключается во время своего выполнения, то ее новые задания не запускаются. При этом уже запущенные продолжают выполняться до конца, после чего очередь останавливается и приобретает статус INACTIVE.

Если отключено задание, то оно не запускается при наступлении всех прочих условий для его запуска. При этом, если не соблюдаются условия для запуска других заданий очереди, она останавливается и приобретает статус INACTIVE. На уже выполняющееся задание его отключение или включение никак не влияет.

Если отключена группа, то данный статус распространяется на все ее задания.

Признак **enabled** имеет приоритет над признаками **bypass** и **skip**. Если задание отключено, пропущено оно не будет.

10. Мониторинг и журналирование

10.1. Логирование сообщений

Логирование ошибочных и информационных сообщений в системе осуществляется средствами пакета LOG_PKG. Запись в лог идет под кодом приложения QMAN. Соответственно шаблон запроса для выборки из лога сообщений системы может быть следующим:

```
select * from logs where app_code = 'QMAN' order by msg_seq desc;
```

10.2. Рассылка диагностических сообщений

В ситуациях, когда требуется вмешательство администратора и система может это отследить, производится рассылка почтовых сообщений по каналу оповещения QMAN. На данный момент сообщения рассылаются в двух случаях:

- когда при запуске очереди обнаруживается, что в ее составе присутствуют отключенные задания либо группы – что рано или поздно приведет к останову очереди в статусе INACTIVE;
- когда очередь переходит в статус, отличный от состояния ожидания либо выполнения.

Администраторам следует подписаться на канал оповещения QMAN для получения подобных сообщений от очередей.

10.3. Журналирование выполнения очередей и заданий

В системе ведется структурированный журнал выполнения. В нем фиксируются вся информация по запуску и выполнению очередей и заданий, с указанием времени начала, окончания и состояния выполнения.

Данные заносятся в единую таблицу как для очередей, так и для заданий. Однако для выборок из журналов следует пользоваться специальными представлениями. Для очередей – QMAN_QUEUE_LOGS_VW, для заданий – QMAN_TASK_LOGS_VW. Выборки напрямую из таблицы будут возвращать неверный текущий статус очередей и заданий.

Структура журналов следующая:

Поле	Описание	Журнал очередей	Журнал заданий
ID	ID журнальной записи	Да	Да
RUN_ID	Уникальный номер запуска	Да	Да
QUEUE_ID	ID очереди	Да	Да
GROUP_ID	ID группы заданий		Да
TASK_ID	ID задания		Да
QUEUE_CODE	Код очереди	Да	Да
GROUP_CODE	Код группы заданий		Да
TASK_CODE	Код задания		Да
RUN_PROC	Исполняемая процедура задания		Да
BYPASS	Признак пропуска выполнения задания		Да
START_DATE	Дата запуска	Да	Да
END_DATE	Дата окончания	Да	Да
STATE	Состояние выполнения	Да	Да
SESSIONID	Уникальный ID сессии		Да
SID	Оперативный ID сессии		Да

Запись в журнал добавляется при запуске очереди или задания, с фиксацией времени запуска. Если затем производится выборка данной записи из журнала, то ее статус может меняться в течение работы журналируемого объекта. По окончании выполнения запись закрывается, при этом происходит фиксация времени окончания выполнения и конечного статуса.

При каждом запуске нового цикла очереди ей назначается т.н. уникальный номер запуска (**run_id**). Тот же самый **run_id** затем назначается всем заданиям, запускающимся в рамках данной очереди. При выборках из журнала это позволяет затем связывать запуски заданий, происходившие в пределах одного цикла выполнения очереди.

Журнальные записи заданий при их сбое закрываются. При дальнейшем перезапуске задания во время восстановления после сбоя в журнал добавляется новая запись. Если при этом задание было разово пропущено (путем установки признака **skip**), то запись в журнал добавляется со значением 2 в поле BYPASS.

Если у задания стоит признак постоянного пропуска **bypass**, то запись добавляется со значением 1 в поле BYPASS. Если же в данном поле стоит 0 или null, то это означает, что был полноценный запуск исполняемой процедуры задания.

Для журнальных записей со статусом BROKEN дата окончания не определена ввиду того, что данный статус подразумевает останов без какой-либо возможности зафиксировать текущий статус. Это стоит учитывать при сборе статистики и подсчете продолжительности выполнения заданий и очередей.

В отличие от заданий, если очередь была полностью остановлена в результате сбоя каких-либо входящих в нее заданий, то после восстановления ее выполнения новая журнальная запись для нее не добавляется.

В журнале также фиксируются уникальный и оперативный системные идентификаторы сессии БД: уникальный SESSIONID соответствует полю V\$SESSION.AUDSID, оперативный SID – соответственно полю V\$SESSION.SID.

10.4. Состояния выполнения очередей и заданий

Очередь может находиться в одном из следующих состояний:

Код состояния	Описание
OK	Состояние ожидания
RUNNING	Состояние выполнения
PREFAIL	В составе очереди есть задания с ошибочным статусом, однако при этом также присутствуют другие параллельно выполняющиеся задания. С точки зрения общего состояния очереди данный статус обозначает пред-ошибочное состояние: очередь еще не остановлена, но если ситуацию по сбойным заданиям в ближайшее время не исправить, то очередь перейдет в статус FAILURE. После устранения ошибок по всем сбойным заданиям очередь снова переходит в статус RUNNING.
FAILURE	В составе очереди есть задания с ошибочным статусом, других параллельно выполняющихся заданий нет. Очередь остановлена по ошибке.
INACTIVE	В составе запущенной очереди нет выполняющихся заданий. Возможная причина – некоторые задания отключены, т.к. сброшен признак enabled . Такие задания как сами не могут выполняться, так и не дают запуститься зависимым (либо расположенным вслед за ними) заданиям.

Задание может находиться в одном из следующих состояний:

Код состояния	Описание
OK	Состояние ожидания
RUNNING	Состояние выполнения
FAILURE	Задание прервано по ошибке. В отличие от статуса BROKEN, данный статус означает корректное закрытие статуса. Фиксация времени окончания выполнения, запись в журналы и пр. происходит в штатном режиме.
BROKEN	Задание прервано ВНЕЗАПНО. Задание может приобрести данный статус в том случае, когда его сессия была принудительно прервана, либо в момент его выполнения была остановлена база. В таком случае задание не имеет возможности зафиксировать ни свой статус, ни время окончания выполнения. Поэтому продолжительность прерванного таким образом задания всегда есть величина неопределенная, в отличие от статуса FAILURE.

Не следует полагаться на статус объектов, полученный прямой выборкой из таблиц QMAN_QUEUES, QMAN_TASKS и QMAN_RUN_LOGS. Он будет иметь неточное значение, не учитывающее текущую динамическую ситуацию. Вместо этого следует пользоваться представлениями QMAN_QUEUES_VW, QMAN_TASKS_VW, QMAN_QUEUE_LOGS_VW и QMAN_TASK_LOGS_VW.

Для получения точного статуса указанные выше представления используют следующие функции:

Процедура	Описание
QMAN_PKG.GET_TASK_STATE	Получить точное состояние задания
QMAN_PKG.GET_QUEUE_STATE	Получить точное состояние очереди
QMAN_PKG.GET_LOG_STATE	Получить точное состояние задания/очереди в журнале выполнения

11. Общая техническая информация

11.1. Перечень составляющих объектов системы

Наименование	Тип	Описание
QMAN_PKG	PACKAGE	Средства управления механизма очередей
QMAN_RUN_PKG	PACKAGE	Ядро механизма очередей
QMAN_QUEUES	TABLE	Очереди заданий
QMAN_GROUPS	TABLE	Группы заданий
QMAN_TASKS	TABLE	Задания
QMAN_DEPEND	TABLE	Зависимости заданий
QMAN_RUN_LOG	TABLE	Журнал выполнения очередей и заданий
QMAN_QUEUES_VW	VIEW	Очереди заданий с точным состоянием выполнения
QMAN_TASKS_VW	VIEW	Задания с точным состоянием выполнения
QMAN_QUEUE_LOGS_VW	VIEW	Журнал выполнения очередей
QMAN_TASK_LOGS_VW	VIEW	Журнал выполнения заданий
QMAN_QUEUES_SEQ	SEQUENCE	Генерация ID для таблицы QMAN_QUEUES
QMAN_GROUPS_SEQ	SEQUENCE	Генерация ID для таблицы QMAN_GROUPS
QMAN_TASKS_SEQ	SEQUENCE	Генерация ID для таблицы QMAN_TASKS
QMAN_DEPEND_SEQ	SEQUENCE	Генерация ID для таблицы QMAN_DEPEND
QMAN_RUN_LOG_SEQ	SEQUENCE	Генерация ID для таблицы QMAN_RUN_LOG
QMAN_RUN_SEQ	SEQUENCE	Генерация ID запусков
QMAN_QUEUE_%	SCHEDULER JOB	Проверка условий и запуск очередей
QMAN_TASK_%	SCHEDULER JOB	Выполнение заданий

11.2. Перечень внешних связанных объектов

Наименование	Тип	Назначение
LOG_PKG	PACKAGE	Логирование
NOTIFY_PKG	PACKAGE	Рассылка сообщений

Кроме указанных выше, для настроенных очередей в список также следует добавить следующие косвенные прикладные связи:

1) Исполняемые PL/SQL процедуры заданий:

```
select run_proc from qman_tasks;
```

2) Процедуры проверки условий запуска очередей:

```
select check_proc from qman_queues where check_proc is not null;
```

11.3. Джобы, входящие в состав системы

Автоматический запуск очередей основан на шедулер-джобах с наименованием в виде QMAN_QUEUE_[QUEUE_ID]. Прямое назначение этих джобов – осуществлять проверку условий запуска очереди, и соответственно запускать их при наступлении этих самых условий. Дополнительное назначение – «подталкивать» уже выполняющиеся очереди, обрабатывая достаточно редкие ситуации, когда лимит одновременно выполняющихся заданий в очереди задействован не полностью.

Задания выполняются в шедулер-джобах с наименованием вида QMAN_TASK_[TASK_ID]. Эти джобы создаются при запуске конкретных заданий и автоматически удаляются по окончании их выполнения.