

ДИНАМИКА, БАЛЛИСТИКА, УПРАВЛЕНИЕ ДВИЖЕНИЕМ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

УДК 629.78

БАЗА ДАННЫХ ПРОГРАММНОГО КОМПЛЕКСА АВТОМАТИЗИРОВАННОГО ПЛАНИРОВАНИЯ ПОЛЕТА АВТОМАТИЧЕСКИХ КОСМИЧЕСКИХ АППАРАТОВ

Жигастова О.К.* , Почукаев В.Н.

*Центральный научно-исследовательский институт машиностроения,
ЦНИИмаш, ул. Пионерская, 4, Королёв, Московская область, 141070, Россия*

** e-mail: inolga-ok@yandex.ru*

Важнейшим элементом программного комплекса автоматизированного планирования полёта автоматических космических аппаратов (КА) [1] является база данных (БД). Она представляет собой инфологическую модель программного комплекса планирования полёта, которая объединяет его элементы в единую систему. БД решает задачи организации работы комплекса, обеспечивает связь между программными модулями, доступ, хранение и управление данными, ведение архива, взаимодействие с комплексами Центра управления полётами (ЦУП) и внешними организациями, участвующими в процессе управления КА. Предложенная структура БД позволила упорядочить хранимые в ней данные, выстроить их иерархию, установить логические связи, обеспечить наглядное представление данных. Благодаря созданной БД была значительно упрощена работа с данными, сокращено время их поиска. Это позволило снизить трудоёмкость и увеличить быстродействие комплекса. БД в составе программного комплекса была внедрена при управлении полётом пятью автоматическими КА и показала свою эффективность в штатных и нештатных ситуациях.

Ключевые слова: автоматический космический аппарат, автоматизированный программный комплекс планирования полёта, база данных автоматизированного программного комплекса.

Для организации, хранения и управления данными, формируемыми программным комплексом автоматизированного планирования полёта автоматических КА, а также для обеспечения внешнего обмена данными с другими комплексами ЦУП была создана база данных программного комплекса планирования полёта. База данных размещена на двух серверах, входящих в состав аппаратно-программных средств ЦУП, объединённых в кластер серверов. В качестве системы управления базой данных (СУБД) выбрана Oracle Server 10g.

Oracle Server 10g является системой управления реляционной базой данных [2]. В реляционной БД данные представляются в виде строк и столбцов таблицы. Построение реляционной БД требует, чтобы в каждой строке таблицы был хотя бы один столбец, значение которого было бы уникальным для этой строки. Уникальность значения гарантирует, что доступ к строке или ее изменение будет производиться независимо от других строк таблицы. Наличие уникальных значений строк обеспечивает поддержку целостности данных БД. Если

модифицировать или удалить строку, имеющую уникальное значение, из одной таблицы БД, то же произойдет с соответствующей строкой в другой таблице, в которой есть ссылка на это значение. В реляционной БД столбец, однозначно определяющий строки таблицы, называется ключевым. При доступе к строкам таблицы БД ключевое значение столбца используется в качестве идентификатора строки.

В реляционных БД выполнение действия по добавлению к таблице новой строки или столбца осуществляется довольно просто. Это позволяет легко устанавливать связи с другими таблицами. В отличие от иерархических или сетевых баз данных, при добавлении нового столбца к таблице реляционной БД не придется переделывать структуру базы, а потребуются внести изменения только в таблицу. Для модификации таблиц и данных, содержащихся в таблицах, можно использовать инструменты СУБД Oracle или операторы языка SQL.

Oracle является распределенной системой обработки данных, для которой может быть организован одновременный доступ сразу с нескольких автоматизированных рабочих мест (АРМ) программного комплекса автоматизированного планирования полёта и комплексов ЦУП. При этом во время взаимодействия с БД на каждом АРМ одновременно может производиться одно или несколько действий по хранению, доступу или обработке данных. Организация работы с данными БД в Oracle реализована по «клиент-серверной» технологии. Подключение АРМ к БД осуществляется с помощью специальных клиентов Oracle, установленных на АРМ. В качестве ответной, «серверной» части используется сама СУБД Oracle. Таким образом, все задачи по обработке данных, реализуемые с помощью СУБД Oracle, можно разделить на две части. К первой части относятся задачи, выполняемые на АРМ, которое отвечает за представление данных специалисту и выполнение каких-либо действий над данными, содержащимися в БД. Для работы с БД в программном комплексе автоматизированного планирования полёта используется язык запросов SQL. С его помощью выполняются все действия над данными БД и показывается результат. Ко второй части относятся задачи, решаемые на «сервере», выполняемые без вмешательства человека. «Серверная» часть ориентирована на выполнение задач «клиента», которые логически разделяются и реализуются компонентами СУБД.

Рассмотрим архитектуру базы данных [2], представленную на рис. 1.

Объектами БД являются таблицы, столбцы, индексы, представления, ограничения, правила, зна-



Рис. 1. Архитектура реляционной БД

чения по умолчанию, триггеры, хранимые процедуры, функции, определяемые пользователем, и типы данных.

Таблицы — это основной объект БД. В них находятся все записи, хранящиеся в базе данных. Каждая строка таблицы содержит столбцы, для которых указываются имя, тип данных и размер.

Индексы используются для уменьшения времени выполнения операций поиска и выборки данных из таблицы. Если в таблице, хранящей большое количество записей, не заданы индексы, то для выборки данных придется полностью просматривать все записи таблицы. Это сильно снижает производительность БД. Индексы создаются на основе одного или нескольких столбцов таблицы и уникально идентифицируют место размещения каждой записи.

Представления используются для выборки некоторого подмножества строк или столбцов из таблицы, объединения столбцов различных таблиц, вычисления итоговых значений на основе данных одного или нескольких столбцов. В БД представление выглядит как обычная таблица, хотя на самом деле оно не содержит данных, а реализуется с помощью операторов языка SQL, которые производят выборку данных из одной или нескольких таблиц.

Ограничения накладываются на значение, которое вводится в столбец. Они добавляются к созданной таблице и могут быть определены как для отдельных столбцов таблицы, так и для самой таблицы.

Правила позволяют ограничивать значения вводимых данных, основываясь на списке значений или логическом выражении, определяющем условие, накладываемое на значение. В отличие от ограничения, правило для столбца таблицы можно определять только одно.

Значение по умолчанию определяет значение, которое автоматически заносится в ячейку столбца, если в нее не было введено другое значение.

Триггер содержит набор инструкций, выполняемых при изменении данных таблицы. Действия триггера зависят от вводимых значений. Он автоматически вызывается во время модификации, добавления или удаления записей таблицы.

Хранимые процедуры используются для реализации различных действий по управлению и обработке данных в таблицах БД. Хранимая процедура представляет собой несколько последовательных инструкций, объединенных в функцию. Она может возвращать значения отдельных параметров и наборы данных. Хранимые процедуры выполняются СУБД.

Физически база данных представляет собой каталогизированную структуру, которая состоит из файлов операционной системы, установленной на кластере серверов, и СУБД, в которой содержатся управляющие, журнальные файлы и файлы данных.

Управляющие файлы содержат информацию о физической структуре БД и управляющую информацию, необходимую для запуска, ведения и восстановления базы. В БД программного комплекса автоматизированного планирования полёта автоматических КА в качестве таких файлов используются файлы control01.ctr, control02.ctr, control03.ctr.

Журнальные файлы являются частью структур восстановления БД и содержат журналы транзакций, в которые записываются все изменения базы. В качестве журнальных файлов в БД программного комплекса автоматизированного планирования полёта используются redo01.log, redo02.log, redo03.log.

Файлы данных содержат в себе все данные БД. В них хранятся логические структуры, такие, как таблицы, кластеры, сегменты отката, индексы и другие объекты БД. В качестве файлов данных используются файлы с расширением .dbf.

Структурно БД разделена на логические области, называемые табличными пространствами. Каждое табличное пространство содержит в себе

объекты БД, они могут размещаться в одном и более файлах данных.

БД программного комплекса автоматизированного планирования полёта автоматических КА имеет имя «SPUTNIK» и состоит из следующих табличных пространств: SYSTEM, INDEXS, TEMP, TOOLS, USERS, AUDIT, EXCH, SBI и PLAN. Каждое табличное пространство содержит по несколько таблиц. Всего в БД программного комплекса автоматизированного планирования полёта более 50 таблиц.

Основной частью табличного пространства SYSTEM является словарь данных. Словарь данных представляет собой набор таблиц и представлений, которые используются для ссылок, обеспечивающих информацию о БД. Словарь данных содержит:

- имена пользователей БД;
- привилегии и роли, выданные каждому пользователю;
- имена и определения объектов схемы;
- ограничения целостности;
- распределения пространства для объектов БД;
- общую структуру БД;
- информацию протоколирования;
- описание триггеров, функций, пакетов и хранимых процедур.

Табличное пространство INDEXS содержит данные по индексам.

Табличное пространство TEMP используется для хранения временных сегментов, которые необходимы для сортировки.

Табличное пространство TOOLS содержит таблицы, необходимые инструментальным средствам Oracle.

В табличном пространстве USERS содержатся следующие пользователи БД: SYS, SYSTEM, USR, SBI, PLAN, OBMEN.

Табличное пространство AUDIT содержит таблицы с данными, обмениваемыми между программным комплексом автоматизированного планирования полета и комплексами ЦУП, а также данные о типе информации и об абонентах, с которыми осуществляется обмен. В таблицах AUDIT содержатся:

- IP адреса АРМ специалистов ЦУП;
- типы хранимой информации;
- сама информация, хранимая в виде файлов данных, включающая табличную и текстовую информацию, формируемую программным комплексом автоматизированного планирования полета, цифровые массивы данных, передаваемых на КА, данные, получаемые с КА, отчёты о проведении сеансов связи и другую полётную информацию.

Табличное пространство EXCH содержит таблицы, необходимые для выполнения программ информационного взаимодействия программного комплекса автоматизированного планирования полета с комплексами ЦУП и сторонними организациями, принимающими участие в управлении КА. В таблицах содержатся:

- типы обмениваемых файлов;
- файлы входной/выходной полетной информации;
- файлы обратных квитанций и коды ответов на передаваемую информацию.

Табличное пространство SBI содержит таблицы, необходимые для хранения баллистической информации, включающей данные о пространственно-временном состоянии КА и получаемой из комплекса навигационно-баллистического обеспечения полета. В таблицах содержатся:

- начальные условия движения КА;
- прогноз параметров орбиты КА;
- интервалы теней и терминаторов;
- интервалы зон радиосвязи командно-измерительных пунктов (КИП);
- файлы измерений текущих навигационных параметров.

Табличное пространство PLAN содержит таблицы, необходимые для выполнения программ комплекса автоматизированного планирования полета. Они содержат:

- общие сведения о КА;
- данные о составе аппаратуры КА и режимах ее работы, выполняемых операциях управления, командных конструкциях [3], командах управления;
- данные о составе средств наземного автоматизированного комплекса управления (НАКУ);
- данные моделирования работы бортового комплекса управления (БКУ) КА;
- архивную информацию, включающую сформированные планы полета КА;
- настроечную информацию, используемую в работе ПО программного комплекса.

В БД для пользователей USERS были созданы следующие объекты:

- таблицы для хранения данных программного комплекса автоматизированного планирования полета автоматических КА и данных, обмениваемых с комплексами ЦУП;
- индексы для быстрого поиска необходимой информации в архивных и обменных таблицах;
- представления для отображения пространственно-временного состояния КА;
- ограничения и правила, определяющие возможные значения для ввода исходных данных при составлении плана полета;

— значения по умолчанию, используемые при моделировании работы БКУ КА;

— триггеры и хранимые процедуры, используемые для чтения и записи в связанные таблицы отдельных значений параметров и файлов, формируемых ПО программного комплекса автоматизированного планирования полета автоматических КА и комплексами ЦУП.

Все данные, хранящиеся в таблицах БД, предназначенные для работы ПО программного комплекса автоматизированного планирования полета автоматических КА, были разделены на четыре вида и размещены каждый в своем табличном пространстве: AUDIT, EXCH, SBI и PLAN.

Общая структура БД программного комплекса автоматизированного планирования полета автоматических КА представлена виде ER-диаграммы на рис. 2. Диаграмма содержит названия таблиц, основные поля данных, ключевые поля, а также связи между таблицами.

Для каждого представленного табличного пространства БД была определена своя структура организации и хранения данных, т.е. должна быть выстроена иерархия таблиц. Рассмотрим каждую структуру.

Для табличного пространства SBI была построена двухуровневая структура. На первом уровне иерархии находится таблица SBI_NU, она содержит начальные условия движения КА. Данная таблица является заголовочной таблицей. Таблица SBI_NU связывает пять таблиц: SBI_BF_100, SBI_PRG, SBI_ITT, SBI_ZRV, SBI_ANOMAL. В них хранятся данные о пространственно-временном состоянии КА, полученные в результате прогнозирования движения КА. Данные таблицы размещены на втором уровне иерархии. В таблице SBI_BF_100 содержится вектор начальных условий движения КА, от которых был произведен расчет для таблиц SBI_PRG, SBI_ITT, SBI_ZRV, SBI_ANOMAL. В таблице SBI_PRG хранятся данные прогноза параметров движения центра масс КА, в таблице SBI_ITT — параметры светотеневой обстановки орбиты, в таблице SBI_ZRV — зоны радиосвязи КИП, в таблице SBI_ANOMAL — зоны геомагнитных аномалий Земли. Для того чтобы получить таблицу, содержащую полный набор параметров пространственно-временного состояния КА, используются представления. Представление SBI_ORBITA обеспечивает отображение данных, характеризующих орбиту КА. Представление SBI_ACTION отображает времена пролета КА над заданными координатами и зонами.

Кроме перечисленных таблиц, в SBI содержатся отдельные не связанные с SBI_NU таблицы, ис-

пользуемые для обработки и хранения измерений текущих навигационных параметров, получаемых с КИП во время проведения сеансов связи с КА. Они представляют собой архив для хранения файлов данных SBI_ITNP и SBI_FILE. SBI_ITNP является заголовочной таблицей для SBI_FILE. Запись и чтение файлов измерений текущих навигационных параметров осуществляется при помощи двух хранимых процедур: SBI_WFILE и RSBI_FILE. Для связи таблиц с таблицей SBI_NU используется таблица SBI_OPERATIONS, через нее осуществляется обмен данными с комплексом навигационно-баллистического обеспечения полета КА.

Табличное пространство PLAN имеет многоуровневую структуру. На первом уровне находится таблица PLAN_KA. Она является заголовочной таблицей для всех таблиц табличного пространства PLAN. В таблице PLAN_KA содержатся общие сведения о космическом аппарате, такие, как его номер, тип, название, дата запуска, срок эксплуатации и др. На втором уровне иерархии находятся три вида таблиц:

1. Таблицы, предназначенные для хранения технологических данных КА и НАКУ, используемых для составления плана полета.

2. Таблицы, предназначенные для хранения данных о состоянии БКУ КА, используемые при моделировании работы БКУ с целью проверки правильности составления плана полета, передаваемого на КА.

3. Настраиваемые таблицы, предназначенные для хранения данных о работе ПО программного комплекса автоматизированного планирования полета автоматических КА.

К таблицам, содержащим данные о КА, относятся таблицы, содержащие данные о составе аппаратуры КА PLAN_TABINI и командных конструкций [3]: режимные операции управления PLAN_REGIME, простые и групповые операции PLAN_TO, циклограммы PLAN_CK и макрокоманды PLAN_MACRO, таблицы разовых PLAN_RK и программных PLAN_PK команд управления. Это простые таблицы, в которых содержатся списочные данные. Они не имеют заголовочных таблиц и связаны напрямую с общей таблицей PLAN_KA. Кроме связи с PLAN_KA, таблица PLAN_TABINI также имеет связь с указанными таблицами через названия систем и устройств КА, для управления которыми используются командные конструкции и команды.

Для записи и чтения тел командных конструкций [4] в БД были созданы восемь хранимых процедур, по две на каждую таблицу:

PLAN_WREGIME и PLAN_RREGIME,
PLAN_WTO и PLAN_RTO,
PLAN_WCK и PLAN_RCK,
PLAN_WMACRO и PLAN_RMACRO.

Они используются ПО формирования плана полета программного комплекса автоматизированного планирования полета автоматических КА.

Таблицы, содержащие данные о НАКУ, имеют трехуровневую структуру. На первом уровне находится таблица PLAN_KIP, в которой содержатся общие данные о составе КИП, используемых для управления автоматическими КА, входящих в НАКУ. В таблице указаны номер КИП, его название и место расположения. На втором уровне для каждого КИП в таблице PLAN_KIPMEANS содержится состав средств, привлекаемых для управления КА. В таблице указывается номер средства и дается описание, для чего оно используется: для передачи командно-программной информации, приема телеметрической информации или задействования каналов линий связи. Таблица PLAN_KIPMEANS связана с таблицей PLAN_KIP через ключевой столбец, содержащий номер КИП. На третьем уровне в таблице PLAN_KIPREGIMS содержится состав режимов средств КИП. В таблице указывается номер режима и дается описание, для выполнения какой операции, проводимой в сеансе связи с КА, он предназначен. Таблица PLAN_KIPREGIMS связана с таблицей PLAN_KIPREGIMS через ключевой столбец, содержащий номер средства КИП. Все три таблицы НАКУ связаны с таблицей PLAN_KA через таблицу PLAN_KIPKA, которая объединяет данные этих таблиц через номера КА, обслуживаемых данными КИП.

Таблицы, предназначенные для хранения данных о состоянии БКУ КА, имеют трехуровневую структуру. На первом уровне находится одна таблица PLAN_ASSV. Она содержит два вида данных: это исходные и смоделированные параметры расчетной орбиты КА. Таблица PLAN_ASSV является заголовочной таблицей для таблиц

PLAN_MODEL_PARAM_VITOK,
PLAN_MODEL_PARAM_CK,
PLAN_MODEL,

содержащих, так же как и PLAN_ASSV, исходные параметры, предназначенные для вычисления состава выдаваемых БКУ в соответствии с планом полета программных команд управления и смоделированные программные команды управления, реализованные БКУ. Перечисленные таблицы связаны между собой номером витка, для которого производятся вычисления.

Таблица `PLAN_MODEL_PARAM_CK` является заголовочной таблицей для таблицы `PLAN_MODEL_ZCK`. Она содержит план выполнения БКУ программных команд управления, входящих в состав циклограмм. Таблица `PLAN_MODEL_ZCK` имеет связи с двумя таблицами: таблицей `PLAN_MODEL_PARAM_CK` с номером циклограммы и таблицей `PLAN_ASSV` с номером витка. Также на этом уровне находится еще одна таблица `PLAN_MODEL_ZPK`, тоже содержащая план выполнения БКУ программных команд, не входящих в состав циклограмм. Она имеет связь только с таблицей `PLAN_ASSV`.

К настроечным таблицам относятся:

`PLAN_ASSVSETUP`, `PLAN_MODELSETUP`,
`PLAN_OPPSETUP`, `PLAN_OBMENSETUP`,
`PLAN_MDT`, `PLAN_PSSSETUP`,
`PLAN_PZSSETUP`, `PLAN_SPPSETUP`,
`PLAN_VPSETUP`.

Все перечисленные таблицы содержат параметры запуска ПО, набор включаемых функций и параметров, отвечающих за формирование пользовательского интерфейса, индивидуально настраиваемого в соответствии с решаемыми задачами для каждого типа КА, шаблоны документов со списком согласующих лиц. Одна из таблиц `PLAN_PSSSETUP` является заголовочной таблицей `PLAN_MDT`, в которой содержатся индивидуальные настройки графического отображения [1] суточного плана полета на экране монитора. Чтение и запись шаблонов документов выполняется с использованием 18 хранимых процедур. Все таблицы, кроме `PLAN_MDT`, имеют связь с общей таблицей `PLAN_KA` через номер КА.

На третьем уровне иерархии находятся таблицы, предназначенные для ввода исходных данных с использованием ПО программного комплекса автоматизированного планирования полета автоматических КА. Ввод исходных данных для работы ПО можно осуществлять двумя способами: с использованием файлов данных, составленных при помощи языка разработки плана полета [4], и с помощью специальных таблиц. Их всего пять:

`PLAN_OPPARRAY`,
`PLAN_PSSARRAY`,
`PLAN_PZSARRAY`,
`PLAN_SPPARRAY`,
`PLAN_TDSSARRAY`.

Заголовочными таблицами для них являются соответствующие настроечные таблицы.

Рассмотрим процедуру ввода исходных данных с использованием указанных таблиц. Структура каждой таблицы ввода исходных данных представ-

ляет собой набор столбцов, которые связаны со столбцами других таблиц, содержащих данные о пространственно-временном состоянии КА и технологические данные КА и НАКУ. Специалист, используя ПО программного комплекса автоматизированного планирования полета, осуществляет последовательный выбор из предложенных ему данных в столбцах таблицы, постепенно заполняя всю таблицу. Ввод специалистом некорректных данных контролируется при помощи ограничений и правил. Если специалист попытается ввести некорректное значение, то оно будет немедленно отвергнуто. Количество столбцов и характер вводимых в них данных устанавливаются в заголовочной таблице. По окончании работы специалиста с ПО данные таблицы ввода, если они не были сохранены, будут потеряны, поскольку при каждом новом запуске ПО выполняется процедура по очистке таблицы ввода.

Кроме названных таблиц, на этом уровне иерархии находится еще одна таблица — `PLAN_DATA`. Это архивная таблица, предназначенная для хранения файлов данных незавершенных планов полетов или просто для хранения типовых планов, которые можно загружать в таблицы ввода данных и дорабатывать. По завершении работы специалист может сохранить незавершенный план в таблице `PLAN_DATA`. Для этого с помощью ПО введенные данные преобразуются в файл, который затем записывается в таблицу `PLAN_DATA` хранимой процедурой `PLAN_WDATA`. Кроме записывающей процедуры, есть хранимая процедура, используемая для чтения файла `PLAN_RDATA`. С ее помощью выполняется обратная операция: файл с незавершенным планом берется из таблицы и преобразуется с помощью ПО в данные, которые помещаются в таблицу ввода.

Табличное пространство `AUDIT` представляет собой каталог хранимой информации. В него записываются цифровые массивы информации, сформированные документы и отчеты с целью их передачи комплексам ЦУП. Каталог организован с помощью таблицы `AUDIT_ARCHIVE`. Она содержит один ключевой столбец с номером хранимой записи и содержимое файла данных. Каждый файл этого каталога имеет два идентификатора: тип хранимой информации и IP-адрес АРМ, на котором он был сформирован. Данные идентификаторы предназначены для облегчения поиска нужного файла в каталоге путем сужения области поиска. Тип и IP-адрес находятся в связанных с `AUDIT_ARCHIVE` таблицах `AUDIT_FILE_TYPE` и `AUDIT_USERS`.

Для организации быстрого поиска записей в таблице `AUDIT_ARCHIVE` были введены первич-

ные индексы. Индексы расставлялись на дату записи файлов в таблицу AUDIT_ARCHIVE. Такая организация позволяет обрабатывать записи таблицы как последовательно, так и выборочно, с произвольным доступом, осуществляемым по заданному значению индекса.

Для записи и чтения файлов каталога были сделаны две хранимые процедуры: AUDIT_WDATA и AUDIT_RDATA. Они возвращают значение полученного номера записи в каталоге. Хранимые процедуры используются ПО обмена и ПО формирования плана полета программного комплекса автоматизированного планирования полета автоматических КА для чтения и записи формируемых файлов.

Табличное пространство EXCH, как и AUDIT, представляет собой каталог обмениваемой с комплексами ЦУП информации. Файлы данных из таблицы AUDIT_ARCHIVE с помощью ПО обмена программного комплекса автоматизированного планирования полета упаковываются в транспортные файлы, предназначенные для обмена информацией по специальным протоколам с комплексами ЦУП и внешними организациями, участвующими в процессе управления КА. Эти файлы помещаются в каталог с целью их выдачи абонентам. Каталог организован с помощью двух таблиц: EXCH_BUFFER, содержащей транспортные файлы, и EXCH_ACK, содержащей файлы ответных квитанции. В EXCH_BUFFER определен ключевой столбец с номером записи передаваемого файла, он имеет связь со столбцом в таблице EXCH_ACK. Как только в таблице EXCH_ACK появляется новая запись с файлом ответной квитанции, она автоматически связывается с записью исходного файла.

Так же как и в AUDIT, для записи и чтения файлов в EXCH используются по две хранимые процедуры на запись и на чтение для каждой из таблиц каталога. Для таблицы EXCH_BUFFER это

EXCH_WFILE и EXCH_RFILE,

а для EXCH_ACK это

EXCH_WACK и EXCH_RACK.

Хранимые процедуры используются ПО обмена программного комплекса автоматизированного планирования полета автоматических КА для чтения и записи передаваемых файлов.

Для поиска нужных файлов и их квитанций в каталоге определен один идентификатор — тип передаваемых данных. Список всех типов обмениваемых файлов содержится в таблице EXCH_TYPE. К ней идут две связи от таблиц EXCH_BUFFER и EXCH_ACK.

Для протоколирования событий обмена данными через EXCH_BUFFER была создана таблица EXCH_PROTOCOL. Запись данных в нее осуществляется с использованием триггеров. Триггер EXCH_ADDFILE добавляет новую запись в таблицу протокола о выдаче хранимого файла абоненту, который запускается по свершении события добавления новой строки в EXCH_BUFFER. Другой триггер, EXCH_ADDACK, запускается по свершении события добавления новой строки в EXCH_ACK. С помощью него в таблицу протокола добавляется запись о получении квитанции от абонента на переданный файл.

Выводы

1. Созданная база данных является связующим элементом программного комплекса автоматизированного планирования полета автоматических КА. С ее помощью осуществляются доступ и хранение данных программного комплекса, а также поддержание их целостности.

2. Рассмотренная архитектура реляционной БД позволила разработать структуру, в которой все данные программного комплекса автоматизированного планирования полета были разделены на табличные пространства, их объектами стали таблицы, выстроенные по уровням иерархии, представления, хранимые процедуры, триггеры и индексы, упростившие процедуру поиска и ввода данных.

3. Использование БД также позволило упростить процедуру организации взаимодействия программных модулей комплекса автоматизированного планирования полета, обеспечив им независимый друг от друга доступ к данным. Благодаря этому существенно упростилось конструирование самого программного комплекса, делая возможным его построение из отдельных независимых блоков.

4. База данных в составе программного комплекса планирования полета была внедрена в ЦУП и использовалась при управлении полетом автоматических КА научного и социально-экономического назначения.

Библиографический список

1. Жигастова О.К., Почукаев В.Н. Программный комплекс автоматизированного планирования полета автоматических космических аппаратов // Вестник Московского авиационного института. 2014. Т. 21. №4. С.60-70.
2. Томас Кайт. Oracle для профессионалов: архитектура, методики программирования и особенности версий 9i, 10g, 11g / Пер. с англ. Н.А. Мухина; под ред. Ю.Н. Артеменко. — 2-е изд. — М.: «Вильямс», 2011. — 848 с.

3. Жигастова О.К., Почукаев В.Н. Командные конструкции и использование их при автоматизированном планировании полёта автоматических космических аппаратов // Вестник Московского авиационного института. 2012. Т.19. №5. С.21-31.
4. Жигастова О.К., Почукаев В.Н. Язык разработки плана полёта, составляемого автоматизированным способом для автоматических космических аппаратов // Вестник Московского авиационного института. 2016. Т.23. №1. С. 136-146.

AUTOMATED FLIGHT PLANNING SOFTWARE COMPLEX FOR UNMANNED SPACECRAFT DATABASE

Zhigastova O.K.* , Pochukayev V.N.

*Central Research Institute of Machine Building,
4, Pionerskaya str., Korolev, Moscow region, 141070, Russia
e-mail: inolga-ok@yandex.ru

Abstract

The article considers the database (DB) designed for software complex of automated flight planning for unmanned spacecrafts (SC) [1]. This DB is meant for storing data for flight planning complex, as well as exchanging data with other complexes of the Mission Control Center (MCC) and the external organizations involved in the spacecraft control process.

The method under consideration for developing a database involves the use of a relational database architecture. A relational database presents all data in rows and columns of the table. Areas of tables' allocation are called table space. Apart from these tables they also contain indexes, notations, constraints, rules, default settings, triggers, stored procedures, user-defined functions and data types. The above listed elements are the objects of the database that are used for its building. The unmanned spacecraft automated flight planning database is structurally divided into four table spaces. The first contains information about the space-time state of the spacecraft. The second contains data necessary to solve the problems of automated planning and its automated checking procedure. The third is the archive of data exchange between software complex of automated flight planning and MCC complexes. The fourth stores output data, transport files and receipt tickets generated in the course of the exchange with the MCC complexes and external organizations involved in spacecraft control process..

All DB tables are arranged in accordance with hierarchical structure. We use two types of tables: table header and their subordinates. The header table contains general information and characteristics of the object, such as the composition of spacecraft groups or ground stations that provide control, while subordinate ones

contain specific data structure of the spacecraft equipment, lists of commands, the types of tools used during the sessions.

The database has been designed to integrate the individual elements of unmanned spacecraft automated flight planning into a unified complex. It presents the link between the complex software modules and provides their interaction with each other and with MCC complexes.

The outcome of the work led to the development of unmanned spacecraft automated flight planning software complex Database. The proposed structure of the database allowed arrange data, set forth its hierarchy, provide storage and access to data and support data integrity.

The created database allowed simplify the separate software modules interrelation procedure of the automated flight planning complex and MCC complexes, providing multistream data access as well as simplifying the search for necessary information.

The developed Database was implemented at MCC as a part of an automated spacecraft flight planning software complex. It saw used in the course of flight control of a spacecraft of scientific and socio-economic purpose.

The automated flight planning for unmanned spacecraft software complex Database can be used for automatic spacecraft of scientific and socio-economic purpose control.

1. The developed database is the binding element of an unmanned spacecraft automated flight planning. It helps to access and store the software complex data and to maintain its integrity as well.

2. This relational database architecture has allowed develop the structure where all automated flight planning

data was divided into table spaces. Their objects are tables formed by levels and hierarchy, notations, stored procedures, triggers and indices simplifying the searching procedure and data entry.

3. Using the database also made it possible to simplify the procedure of interaction of automated flight planning software modules, providing them with data access independently of one another. This significantly simplified the design of the software complex itself, allowing its construction with separate independent blocks.

4. The Database was implemented at the MCC and was used for spacecraft of scientific and socio-economic purpose automated flight control as a part of flight planning complex.

Keywords: automatic spacecraft, automated flight planning software complex, automated software complex database.

References

1. Zhigastova O.K., Pochukayev V.N. *Vestnik Moskovskogo aviatsionnogo instituta*, 2014, vol. 21, no. 4, pp. 60-70.
2. Thomas Kyte. *Expert Oracle Database Architecture: Oracle Database 9i, 10g and 11g Programming Techniques and Solutions*, Apress, 2010, 833 p.
3. Zhigastova O.K., Pochukayev V.N. *Vestnik Moskovskogo aviatsionnogo instituta*, 2012, vol. 19, no. 5, pp. 21-31.
4. Zhigastova O.K., Pochukayev V.N. *Vestnik Moskovskogo aviatsionnogo instituta*, 2016, vol. 23, no. 1, pp. 136-146.