

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального
образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Н.Э. БАУМАНА
(МГТУ им. Н.Э. Баумана)
Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

Брешенков А.В., Волкова Е.А., Галямова Е.В.

Разработка DB2 приложений

Методические указания
для выполнения лабораторной работы по курсу
«СЕТЕВЫЕ БАЗЫ ДАННЫХ»

2009 год,
Москва

Лабораторная работа «Разработка DB2 приложений»

Аннотация

Система управления базами данных СУБД DB2 компании IBM — это мощная высокопроизводительная многоплатформенная СУБД. Она поставляется с целым набором клиентских инструментов, которые могут быть использованы для выполнения операторов языка структурированных запросов SQL.

База данных DB2 для z/OS отличается высокой степенью готовности, масштабируемостью и производительностью при обработке транзакций, в неё включена поддержка очень больших баз данных благодаря поддержке 64-разрядных виртуальных систем хранения. Однако существуют некоторые специфические особенности работы с СУБД на больших ЭВМ класса IBM мэйнфрейм System z, которым и посвящено данное методическое указание.

В данном методическом указании изложены основы разработки приложений для работы с базами данных DB2. Дано краткое описание компиляции программ в операционной среде z/OS, особенностей подготовки программ в DB2, статических и динамических SQL-операторов. В практической части приведены примеры кода программ на языках COBOL и PL/1 и указаны настройки утилит DB2 для подготовки этих программ к запуску.

Цели лабораторной работы:

- Изучение функций DCLGEN;
- Разработка структуры программ, работающих с базами данных;
- Изучение этапов подготовки программы к запуску.

1. Теоретическая часть

1.1 Компиляция программ в z/OS

Функция компилятора состоит в том, чтобы преобразовать исходный код в объектный модуль, который затем должен пройти обработку компоновщиком (или редактором связей), прежде чем его можно будет выполнять. В процессе компиляции исходного модуля компилятор присваивает относительные адреса всем инструкциям, элементам данных и меткам, начиная с нуля. Адреса состоят из базового адреса и смещения. Это позволяет изменять расположение программ; другими словами, их необязательно загружать в одно и то же место памяти при каждом выполнении программ.

Соруbook – разделяемая библиотека, в которой программисты хранят часто используемые сегменты программ

Редактор связей преобразует объектные модули в исполняемые загрузочные модули

Любые ссылки на внешние программы и подпрограммы остаются неразрешенными. Эти ссылки либо разрешаются при компоновке объектного модуля, либо динамически разрешаются при выполнении программы.

Для компиляции программ в z/OS можно использовать пакетное задание или осуществлять компиляцию в TSO/E, используя команды, CLIST-программы или ISPF-панели.

Программы на языке C можно компилировать в оболочке z/OS UNIX командой c89. Программы на языке COBOL можно компилировать в оболочке z/OS UNIX командой cob2.

Для компиляции в пакетном задании z/OS включает набор каталогизированных процедур, позволяющих избежать ввода JCL-кода, который иначе нужно было бы набирать. Если ни одна из каталогизированных процедур не соответствует вашим требованиям,

потребуется написать весь JCL-код для компиляции.

На этапе компиляции нужно определить наборы данных, требуемые для компиляции, любые опции компилятора, необходимые для вашей программы, и желательный формат вывода.

1.2 Программирование приложений для DB2

Язык SQL не является полным языком программирования, однако он необходим для доступа и управления данными в базе данных DB2. Он представляет собой не процедурный язык 4-го поколения (4GL), разработанный в середине 1970-х для использования в DB2. Язык SQL можно либо использовать динамически в интерпретационной программе вроде SPUFI, либо встраивать и компилировать или ассемблировать в составе основного языка. Так как же написать приложение, осуществляющее доступ к данным в DB2? Для этого SQL встраивается в исходный код языка программирования, такого как Java, Smalltalk, REXX, C, C++, COBOL, Fortran, PL/I и высокоуровневый ассемблер.

Существует две категории SQL-операторов, которые можно использовать в программе: статические и динамические.

- **Статические.**

Статическими SQL-операторами называются полные SQL-операторы, записанные внутри исходного кода. В процессе подготовки программы, СУБД DB2 определяет пути доступа для операторов, и они записываются в DB2. Операторы SQL не изменяются от одного запуска программы к другому; при этом используются одни и те же заданные пути доступа, так что подсистеме DB2 не требуется создавать их повторно, что может вызвать дополнительную нагрузку (Примечание. Все SQL-операторы должны иметь путь доступа).

- **Динамические.**

Динамическими SQL-операторами называются SQL-операторы, частично

или полностью неизвестные на момент написания программы. Только после запуска программы подсистема DB2 знает, какие операторы будут использоваться, и способна определить требуемые пути доступа. Пути доступа не сохраняются, так как операторы могут изменяться от одного запуска программы к другому. Примером такой программы является SPUIFI. Утилита SPUIFI в действительности представляет собой приложение, принимающее динамические SQL-операторы. Это те операторы, которые вводятся во входном файле. При каждом запуске SPUIFI, SQL-операторы могут быть разными, поэтому в приложение встраиваются специальные подготовительные SQL-операторы.

1.3 Подготовка программы в DB2: алгоритм

Традиционный процесс подготовки программы, состоящий из компиляции и редактирования связей, при подготовке SQL-кода должен включать дополнительные этапы, так как компиляторы не распознают SQL. Эти этапы, наряду с компиляцией и редактированием связей, можно выполнять через панель DB2I, хотя весь процесс, за исключением вызова DCLGEN, обычно выполняется в одном потоке заданий JCL. При чтении объяснений см. рис. 1.

1.3.1 DCLGEN

Утилита DCLGEN предоставляет способ автоматического генерирования исходных определений для объектов DB2, используемых в программе. Они создаются в разделе библиотеки DCLGEN, которую можно включить в вашу исходную программу. Если ее не включить, необходимо составлять определения вручную. Администратор базы данных DB2 обычно создает их на основе правил компании. На этом этапе необходима работающая система DB2, так как определения берутся из каталога DB2.

1.3.2 Прекомпиляция

Так как компиляторы не могут обрабатывать SQL-код, на этапе прекомпиляции SQL-выражения заключаются в комментарии, подставляя оператор CALL обращения к DB2. Он передает некоторые параметры, например, адреса переменных основного языка (для извлечения данных), номера операторов и (что очень важно) измененную отметку времени, называемую маркером согласованности (consistency token; часто называется просто отметкой времени). На этом этапе не требуется запуск системы DB2; все выполняется без доступа к DB2. Прекомпилятор определяет SQL-код по специальным флагам начала и конца, которые должны быть включены для каждого SQL-оператора. Во всех языках программирования используется одинаковый флаг начала – EXEC SQL. Флаги конца различаются. В языке COBOL используется END-EXEC. (точка), тогда как в С и других языках используется точка с запятой. Ниже приведен пример на языке COBOL:

```
EXEC SQL  
SELECT EMPNO, LASTNAME  
INTO :EMPNO, :LASTNAME  
FROM EMP  
END-EXEC.
```

В этом примере, значения EMPNO и LASTNAME извлекаются в переменные основного языка, перед которыми указывается двоеточие. Переменными основного языка являются переменные, определенные на основном языке (COBOL, PL/1 и т. д.) – языке включающем SQL-код. На этапе вызова DCLGEN также происходит определение этих переменных. В данном случае имя переменной основного языка соответствует имени столбца, что не является обязательным требованием; можно использовать любое имя с типом данных, совместимым с типом данных столбца.

После прекомпиляции программа разделяется на две части.

- Измененный исходный код; представляет первоначальный исходный код, в котором SQL-код заключен в комментарии и замещен операторами CALL.
- Модуль запросов базы данных (database request module, DBRM), который обычно представляет собой раздел библиотеки PDS и содержит SQL-операторы программы.

Измененный исходный код передается компилятору для компиляции и редактирования связей, вследствие чего создается исполняемый загрузочный модуль, как для любой другой программы, не содержащей SQL. Кстати, можно вставить в программу любой тип SQL-кода: DML, DDL и DCL, если только не нарушаются правила авторизации.

1.3.3 Связывание

В DB2 связывание можно рассматривать как аналог процесса компиляции для DBRM.

Связывание выполняет три операции.

- Выполняет проверку синтаксиса на наличие ошибок.
- Выполняет проверку полномочий.
- Что важнее всего, определяет пути доступа для операторов. DB2 содержит компонент, называемый оптимизатором (optimizer), который рассматривает различные способы осуществления доступа к данным, такие как сканирование целой таблицы, использование индекса, выбор используемого индекса и т. д. Он определяет показатели стоимости каждого пути доступа и выбирает путь доступа с наименьшей стоимостью. Такой оптимизатор называется оптимизатором на основе стоимости (в отличие от оптимизатора на основе правил). Код SQL вместе с путем доступа (и маркером согласованности – отметкой времени) хранится как пакет в оглавлении DB2. Другая информация, в частности информация о пакете и действительный SQL-код, хранится в каталоге. При связывании

создается исполняемый SQL-код для одного приложения в пакете. Теперь DB2 имеет всю информацию, необходимую для получения запрашиваемых данных для данной программы. Часто программы вызывают подпрограммы, которые также содержат SQL-вызовы. Каждая из этих подпрограмм также имеет пакет. Необходимо сгруппировать всю информацию DB2 вместе. Для этого необходим еще один этап: еще одно связывание, но на этот раз предназначенное для создания плана (plan). Даже если подпрограмма не используется, все равно требуется создать план. План может содержать не только информацию о программе, но и другую информацию. Это представляет обычную практику: план содержит все пакеты одного проекта, и при каждом запуске используется один и тот же план. Для полноты следует добавить, что изначально DBRM связывались непосредственно в планы (они назывались встроенными DBRM). Однако при внесении небольшого изменения в одну из программ, необходимо выполнить повторное связывание всего плана. Это также необходимо сделать и в случае добавления индекса. Как вы знаете, в процессе связывания DB2 обновляет свои оглавление и каталог. Обновление означает недопущение их обновления другими людьми (данные заблокированы и недоступны для них), так что выполнение большинства других действий в DB2 является практически невозможным. Для того чтобы обойти это ограничение, были реализованы пакеты. Теперь необходимо только лишь выполнить повторное связывание одного пакета, так что длительность обновления очень мала, и воздействие на других пользователей почти отсутствует. В настоящее время все еще используются планы с встроенными DBRM, хотя большинство компаний выполняет их преобразование в пакеты. Планы используются только в среде мэйнфреймов. На других платформах они не используются.

1.3.4 Запуск

При выполнении приложения происходит загрузка загрузочного модуля в основную память. Когда встречается SQL-оператор, обращение к DB2 (оператор CALL), заменивший SQL-оператор, передает его параметры в DB2. Одним из параметров является маркер согласованности. Этот же маркер (или отметка времени) присутствует и в пакете. Затем выполняется просмотр пакетов в заданном плане DB2 в поисках наличия соответствующей отметки времени, после чего загружается и выполняется подходящий пакет. Таким образом, для запуска необходимо в качестве параметра задать имя плана.

Последнее замечание: результатом SQL-выражения обычно является результирующее множество (содержащее больше одной строки). Приложение может обрабатывать только одну запись или строку за раз. В DB2 добавляется специальная конструкция, называемая курсором (cursor), который, в сущности, аналогичен указателю и позволяет во встроенном SQL-коде выбирать, обновлять или удалять по одной строке из результирующего множества за один раз.

2. Практическая часть

В этой лабораторной работе вы будете создавать приложение, обрабатывающее различные запросы к базе данных DB2, в том числе и сетевые запросы. Последовательно выполняя этапы 1-4, вы познакомитесь с основами разработки приложений, работающих с DB2, и последовательностью их запуска.

2.1 DCLGEN

Сначала вы будете использовать DCLGEN для создания SQL DECLARE и основной переменной структуры для таблиц и видов, которые вы только что создали, а также для master таблиц, которые вы будете копировать, все из которых будут доступны в программах, которые вы

будете обновлять. DCLGEN можно запустить из меню DB2I пункт 2. Изучите различные опции на экране. Используйте HELP, если это необходимо.

Теперь используем DCLGEN для EMP таблицы.

1. Заполните опции DCLGEN (см. рис.2), а затем нажмите клавишу "Enter" для запуска:

TSOU_{sxx} – ваш логин.

2. Сделайте то же самое для таблицы DEPT.

3. Сделайте то же самое для вида VPHONE.

4. Сделайте то же самое для таблицы master DEPT, которая вам позже понадобится в программе. Название таблицы master DEPT - CF82.DEPT_MAST.

5. И наконец, сделайте то же самое для CF82.EMP_MAST таблицы. Предложение: Имя набора для CF82.EMP_MAST структуры: 'TSOU_{sxx}.CF82.INCLUDE (EMPMAST)'

***Примечание:** При генерации двух выходных наборов для EMP таблицы и EMP master таблицы, вы увидите, что DCLGEN сгенерировал два одинаковых набора основных переменных внутри структуры. Ваша сборка не удастся (неоднозначные имена переменных), если вы не определите каждое имя переменной при написании кода.*

6. Просмотрите содержимое сгенерированных частей (через ISPF Browse)

7. Вы можете экспериментировать с опциями 8 и 12, чтобы узнать получаемые результаты. Используйте:

ACTION ...====> REPLACE.

2.2 Структура программы

Исследуйте следующие экраны, они показывают транзакции требующие обработки DB2.

Первый экран предлагает несколько функций (см. рис.3)

Представим, что выбрана функция s, появится следующий экран (см. рис.4)

Программа использует введенный номер и предоставит требуемую информацию, а затем вернется опять к первому экрану.

Представим, что выбрана функция a, появится следующий экран (см. рис.5)

Программа использует введенные данные и добавит новые данные в DB2.

Вам предоставляется готовый код программы на языках PL/I или COBOL.

Можно использовать любой язык на выбор. Имя программы LABPGMCS (COBOL) или LABPGMPS (PL/I).

Изучите код программы, используя ISPF Browse.

Обратите внимание на использование операторов SQL, например, EXEC SQL INCLUDE EMP END-EXEC.

1. Найдите слова SQL COMMUNICATION AREA.
2. Изучите SQL выражение добавляющее определение SQLCA в программу.
3. Найдите слова ERROR HANDLING.
4. Изучите SQL для обработки ошибок
5. Найдите слово INDICATOR.
6. Изучите как объявляется переменная используемая DB2 для обозначения значений NULL SELECTED и используемая в программе для установке значений NULL в выражениях INSERT и UPDATE.
7. Найдите в программе блоки S, P, D, I, U, A. Определите выполняемые в них действия.

2.3 Подготовка программы

- Выберите Пункт 3 DB2I для подготовки программы.
- Исследуйте опции на экране, и на всех последующих экранах,

используя HELP, если это необходимо.

- На экране показанном на рисунке 6 - то, что вам нужно для запуска всех необходимых шагов для подготовки программы для запуска.
- Сначала будем отображать все экраны DB2I относящиеся ко различным шагам подготовки, так как мы используем эти функции в первый раз.
- Затем Вы будете заменять большинство из YES на Display панели на NO.
- pgmname во всем руководстве следует заменять на название источника программы.
- имена наборов данных могут варьироваться.

Следующая панель, которую вы должны получить, это PRECOMPILE панель:

1. Заполните панель как показано на рисунке 7.

Следующая панель, которую вы получите, это BIND PACKAGE панель

2. Заполните панель как показано на рисунке 8.
3. Заполните BIND PACKAGE варианты по умолчанию, как показано на рисунке 9. Сделайте то же самое на следующем экране, на COMPILE, PRELINK, LINK и RUN панели (см. рис.10).
4. Нажмите кнопку ENTER и проверьте вывод сообщений с целью проверки выполнения действия.
5. Исследуйте элементы вашего набора данных DBRM (TSOUSxx.CF82.DBRM).
6. Ваша программа подготовки сгенерировала определенные наборы данных:

TSOUSxx.TEMP.COBOL Модифицированный COBOL исходный код

TSOUSxx.TEMP.PLI Модифицированный PL / I исходный код

Связь TSOUSxx.TEMP.DATA выход редактора связей

Связь TSOUSxx.TEMP.LINKLIST редактор связей

TSOUSxx.TEMP.LIST листинг компилятора

Объект TSOUsxx.TEMP.OBJ объектный модуль

TSOUsxx.TEMP.PCLIST листинг препроцессора

Ваша программа не запустится, пока вы не выполните следующей части этого упражнения!

Прежде, чем мы можем запустить программу, мы должны создать план с листингом пакета, который указывает на наш пакет.

1. Составим план с таким же именем, что и программа, и с листингом пакета, выполнив следующее (см. рис. 11) и нажав Enter:

Давайте определим параметры BIND PLAN и составим план.

Следующая панель, которую вы получите, это BIND PLAN панель:

2. Заполните панель как показано на рисунке 12.

Нужно оставить пустыми поля опций 1 и 3. Иначе, вы создадите внутривидеопотоковый DBRM.

3. Заполните настройки BIND PLAN по умолчанию как показано на рисунке 13.

4. Создайте листинг пакета.

DB2I предлагает выбрать листинг пакета для плана (см. рис. 14).

Единственный возможный выбор отображается на экране. Можно также выбрать использование collid .*, но тогда вам потребуется несколько дополнительных привилегий чтобы можно было выполнить это.

Указывая TSOUsxx.pgmname в листинге пакета, Вы выбираете только один пакет, который хранится в вашей коллекции.

Закройте панель нажатием PF клавиши F3.

На данный момент, вы можете запускать программу!

2.4 Запуск программы

Код JCL для запуска вашей программы располагается в наборе данных TSO: CF82.CNTL (RUNPGMC) или CF82.CNTL (RUNPGMP).

Текст скрипта RUNPGMC:

```
//TSOUxxx A JOB , 'DB2 RUN PROGRAM'  
//PH01S01 EXEC PGM=IKJEFT01, DYNAMNBR=20, COND=(4,LT)  
//*TEPLIB DD DSN=DB2.SDSNLOAD, DISP=SHR  
//CARDIN DD DSN=TSOUxxx.CF82.SOURCE(TRANSIN), DISP=SHR  
//REPORT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//SYSABOUT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *  
DSN SYSTEM(DSNC)  
RUN PROGRAM(pgmname) PLAN(pgmname) -  
LIB(RUNLIB.LOAD)  
/*
```

Обратите внимание на следующее:

- Названием программы является IKJEFT01 – Запускается в TSO.
 - имя CARDIN DD -расположение входного файла TRANSIN.
 - имя REPORT DD - вывод - то же место, что и вывод задачи.
 - имя SYSTSIN DD определяет TSO команду запускающую загрузку, - в данном случае DSN, команда для запуска команд DB2 из TSO. Код подсистемы также необходимо указать.
 - В СУБД DB2 команда вызываемая здесь – RUN PROGRAM, с указанием имени плана и имени программы(загрузочного модуля). Библиотека загрузочного модуля также указана. Почему местонахождение плана не нужно указывать?
1. Запустите этот JCL.
 2. Посмотрите на результаты в ISPF 3.8.
 3. Обратите внимание на код возврата.

4. Кроме того, убедитесь в том, что вывод из программы (результаты действий S, P, D, I, U, A из TRANSIN файла) согласован с выводом-образцом.

Важное примечание: Любые изменения, вносимые в исходный код теперь должен быть приведены в действие повторением всех предыдущих шагов подготовки программы!

3. Требования к отчету

Отчет о выполнении лабораторной работы должен содержать ответы на вопросы, поставленные перед студентом в методических указаниях, скрин-шоты результатов компиляции программы, скрин-шоты результатов выполнения разрабатываемой программы и ответы на контрольные вопросы. Оформление отчета должно быть выполнено в соответствии с требованиями ГОСТов.

4. Контрольные вопросы

Как вставить код SQL в программу?

Какие шаги необходимо проделать для подготовки программы DB2 к запуску?

Для чего необходима утилита DCLGEN?

Чем отличается пакет от плана?

5. Список литературы

5.1 *Paolo Bruni, Mark Anders, «DB2 for z/OS: Data Sharing in a Nutshell», International Business Machines Corporation, 2006.*

5.2 *Справочник «DB2 SQL Reference», International Business Machines Corporation, 2003.*

5.3 *Mike Ebbers, Wayne O'Brien, Bill Ogden, «Introduction to the New Mainframe: z/OS Basics», IBM corp., 2005.*

Содержание:

1.	Теоретическая часть	3
1.1	Компиляция программ в z/OS	3
1.2	Программирование приложений для DB2.....	4
1.3	Подготовка программы в DB2: алгоритм.....	5
1.3.1	DCLGEN	5
1.3.2	Прекомпиляция	6
1.3.3	Связывание	7
1.3.4	Запуск	9
2.	Практическая часть.....	9
2.1	DCLGEN	9
2.2	Структура программы	10
2.3	Подготовка программы	11
2.4	Запуск программы.....	13
3.	Требования к отчету	15
4.	Контрольные вопросы	15
5.	Список литературы	15

===>

Enter table name for which declarations are required:

1 SOURCE TABLE NAME ===> EMP

2 TABLE OWNER ===>

3 AT LOCATION ===> (Optional)

Enter destination data set: (Can be sequential or partitioned)

4 DATA SET NAME ... ===> 'TSOU\$xxx.CF82.INCLUDE(EMP)'

5 DATA SET PASSWORD ===> (If password protected)

Enter options as desired:

6 ACTION ===> ADD (ADD new or REPLACE old declaration)

7 COLUMN LABEL ===> NO (Enter YES for column label)

8 STRUCTURE NAME .. ===> (Optional)

9 FIELD NAME PREFIX ===> (Optional)

10 DELIMIT DBCS ===> YES (Enter YES to delimit DBCS identifiers)

11 COLUMN SUFFIX ... ===> NO (Enter YES to append column name)

12 INDICATOR VARS .. ===> NO (Enter YES for indicator variables)

13 RIGHT MARGIN ===> 72 (Enter 72 or 80)

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
 F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

Puc. 2

MAIN TRANSACTION

Pick an Option

- S - Select Employee Name Information
- D - Delete a Department
- I - Insert a Department
- U - Update Employee Phone Number
- A - Add Employee
- F - Display a Department

Puc. 3

EMPLOYEE NAME INFORMATION

Please ENTER the employee's serial number in
the space provided:

Puc. 4

ADD A NEW EMPLOYEE TO STAFF

Please ENTER the employee's information below:

Employee Number: -----

Last Name : -----

First Name : -----

Initial : _

Phone Number : -----

Department : ---

Puc. 5

DB2 PROGRAM PREPARATION

COMMAND ==>

SSID:

Enter the following:

- 1 INPUT DATA SET NAME ==> 'TSOU\$xx.CF82.SOURCE(pgmname)'
- 2 DATA SET NAME QUALIFIER ==> TEMP (For building data set names)
- 3 PREPARATION ENVIRONMENT ==> FOREGROUND (FOREGROUND, BACKGROUND, EDITJCL)
- 4 RUN TIME ENVIRONMENT ... ==> TSO (TSO, CAF, CICS, IMS, RRSAP)
- 5 OTHER DSNH OPTIONS ==>

(Optional DSNH keywords)

Select functions:	Display panel?	Perform function?
6 CHANGE DEFAULTS	==> N (Y/N)	
7 PL/I MACRO PHASE	==> N (Y/N)	==> N (Y/N)
8 PRECOMPILE	==> Y (Y/N)	==> Y (Y/N)
9 CICS COMMAND TRANSLATION		==> N (Y/N)
10 BIND PACKAGE	==> Y (Y/N)	==> Y (Y/N)
11 BIND PLAN	==> N (Y/N)	==> N (Y/N)
12 COMPILE OR ASSEMBLE	==> Y (Y/N)	==> Y (Y/N)
13 PRELINK	==> N (Y/N)	==> N (Y/N)
14 LINK	==> Y (Y/N)	==> Y (Y/N)
15 RUN	==> N (Y/N)	==> N (Y/N)

PRESS: ENTER to process END to save and exit HELP for more information

Puc. 6

PRECOMPILE

COMMAND ==>

SSID:

Enter precompiler data sets:

- 1 INPUT DATA SET ==> 'TSOU\$xx.CF82.SOURCE(pgmname)'
- 2 INCLUDE LIBRARY ... ==> 'TSOU\$xx.CF82.INCLUDE'
- 3 DSNNAME QUALIFIER .. ==> TEMP (For building data set names)
- 4 DBRM DATA SET ==> 'TSOU\$xx.CF82.DBRM'

Enter processing options as desired:

- 5 WHERE TO PRECOMPILE ==> FOREGROUND (FOREGROUND, BACKGROUND, or EDITJCL)
- 6 VERSION ==> (Blank, VERSION, or AUTO)
- 7 OTHER OPTIONS ==>

PRESS: ENTER to process END to cancel HELP for more information

Puc. 7

```

                                BIND PACKAGE                                SSID:
COMMAND ==>

Specify output location and collection names:
 1 LOCATION NAME ..... ==>                                (Defaults to local)
 2 COLLECTION-ID ..... ==> TSOUxxx                          (Required)
Specify Package source (DBRM or COPY):
 3 DBRM:          COPY:          ==> DBRM                    (Specify DBRM or COPY)
 4 MEMBER   or COLLECTION-ID ==> pgmname
 5 PASSWORD or PACKAGE-ID .. ==>
 6 LIBRARY  or VERSION ..... ==> 'TSOUxxx.CF82.DBRM'
                                                (Blank, or COPY version-id)
 7 .....  -- OPTIONS ..... ==>                            (COMPOSITE or COMMAND)
Enter options as desired:
 8 CHANGE CURRENT DEFAULTS? .. ==> YES                      (NO or YES)
 9 ENABLE/DISABLE CONNECTIONS? ==> NO                      (NO or YES)
10 OWNER OF PACKAGE (AUTHID).. ==>                          (Leave blank for primary ID)
11 QUALIFIER ..... ==>                                      (Leave blank for OWNER)
12 ACTION ON PACKAGE ..... ==> ADD                         (ADD or REPLACE)
13 INCLUDE PATH? ..... ==> NO                              (NO or YES)
14 REPLACE VERSION ..... ==>

```

PRESS: ENTER to process END to save and exit HELP for more information

Puc. 8

```

                                DEFAULTS FOR BIND PACKAGE                SSID: DSNB
COMMAND ==>

Change default options as necessary:
 1 ISOLATION LEVEL ..... ==> CS                            (RR, RS, CS, UR, or NC)
 2 VALIDATION TIME ..... ==> BIND                          (RUN or BIND)
 3 RESOURCE RELEASE TIME ... ==> COMMIT                     (COMMIT or DEALLOCATE)
 4 EXPLAIN PATH SELECTION .. ==> NO                        (NO or YES)
 5 DATA CURRENCY ..... ==> NO                             (NO or YES)
 6 PARALLEL DEGREE ..... ==> ANY                           (1 or ANY)
 7 SQLERROR PROCESSING ..... ==> NOPACKAGE                 (NOPACKAGE or CONTINUE)
 8 REOPTIMIZE FOR INPUT VARS ==> NO                         (ALWAYS, NONE, or ONCE)
 9 DEFER PREPARE ..... ==> NO                              (NO or YES)
10 KEEP DYN SQL PAST COMMIT. ==> NO                        (NO or YES)
11 DBPROTOCOL ..... ==>                                    (DRDA or PRIVATE)
12 APPLICATION ENCODING..... ==> _                         (Blank, ASCII, EBCDIC, UNICODE,
                                                or ccsid)
13 OPTIMIZATION HINT ..... ==>                             (Blank or 'hint-id')
14 IMMEDIATE WRITE..... ==>                                (YES, NO)
15 DYNAMIC RULES ..... ==> BIND                           (RUN, BIND, DEFINE,
                                                or INVOKE)

```

Puc. 9

PROGRAM PREP: COMPILE, PRELINK, LINK, AND RUN SSID:
COMMAND ==>

Enter compiler or assembler options:

- 1 INCLUDE LIBRARY ==>
- 2 INCLUDE LIBRARY ==>
- 3 OPTIONS ==>

Enter prelink and linkage editor options:

- 4 INCLUDE LIBRARY ==>
- 5 INCLUDE LIBRARY ==>
- 6 INCLUDE LIBRARY ==>
- 7 LOAD LIBRARY .. ==> 'TSOUxxx.CF82.LOAD'
- 8 PRELINK OPTIONS ==>
- 9 LINK OPTIONS .. ==>

Enter run options:

- 10 PARAMETERS ==>
- 11 SYSIN DATA SET ==> TERM
- 12 SYSPRINT DS ... ==> TERM

PRESS: ENTER to process END to save and exit HELP for more information

Puc. 10

DB2 PROGRAM PREPARATION

COMMAND ==> SSID:

Enter the following:

- 1 INPUT DATA SET NAME ==> 'TSOUxxx.CF82.SOURCE (pgmname)'
- 2 DATA SET NAME QUALIFIER ==> TEMP (For building data set names)
- 3 PREPARATION ENVIRONMENT ==> FOREGROUND (FOREGROUND, BACKGROUND,
EDITJCL)
- 4 RUN TIME ENVIRONMENT ... ==> TSO (TSO, CAF, CICS, IMS, RRSAP)
- 5 OTHER DSNH OPTIONS ==>

(Optional DSNH keywords)

- | Select functions: | Display panel? | Perform function? |
|-----------------------------|----------------|-------------------|
| 6 CHANGE DEFAULTS | ==> N (Y/N) | |
| 7 PL/I MACRO PHASE | ==> N (Y/N) | ==> N (Y/N) |
| 8 PRECOMPILE | ==> N (Y/N) | ==> N (Y/N) |
| 9 CICS COMMAND TRANSLATION | | ==> N (Y/N) |
| 10 BIND PACKAGE | ==> N (Y/N) | ==> N (Y/N) |
| 11 BIND PLAN | ==> Y (Y/N) | ==> Y (Y/N) |
| 12 COMPILE OR ASSEMBLE | ==> N (Y/N) | ==> N (Y/N) |
| 13 PRELINK | ==> N (Y/N) | ==> N (Y/N) |
| 14 LINK | ==> N (Y/N) | ==> N (Y/N) |
| 15 RUN | ==> N (Y/N) | ==> N (Y/N) |

PRESS: ENTER to process END to save and exit HELP for more information

Puc. 11

```

                                BIND PLAN                                SSID:
COMMAND ==>

Enter DBRM data set name(s):
 1 MEMBER ..... ==> >>> blank out <<<
 2 PASSWORD ..... ==>
 3 LIBRARY ..... ==> >>> blank out <<<
 4 ADDITIONAL DBRMS? ..... ==> NO          (YES to include more DBRMs)

Enter options as desired:
 5 PLAN NAME ..... ==> pgmname          (Required to create a Plan)
 6 CHANGE CURRENT DEFAULTS? .. ==> YES   (NO or YES)
 7 ENABLE/DISABLE CONNECTIONS? ==> NO   (NO or YES)
 8 INCLUDE PACKAGE LIST?..... ==> YES   (NO or YES)
 9 OWNER OF PLAN (AUTHID)..... ==>      (Leave blank for your
primaryID)
10 QUALIFIER ..... ==>                  (For Tables, Views, and
aliases)
11 CACHESIZE ..... ==> 0                (Blank, or value 0-4096)
12 ACTION ON PLAN ..... ==> ADD         (REPLACE or ADD)
13 RETAIN EXECUTION AUTHORITY. ==> NO   (YES to retain user list)
14 CURRENT SERVER ..... ==>            (Location name)
15 INCLUDE PATH? ..... ==> NO          (No or YES)

PRESS: ENTER to process   END to save and exit   HELP for more information

```

Puc. 12

```

                                DEFAULTS FOR BIND PLAN                SSID: DSNE
COMMAND ==> █

Change default options as necessary:
 1 ISOLATION LEVEL ..... ==> CS          (RR, RS, CS, or UR)
 2 VALIDATION TIME ..... ==> BIND        (RUN or BIND)
 3 RESOURCE RELEASE TIME ... ==> COMMIT   (COMMIT or DEALLOCATE)
 4 EXPLAIN PATH SELECTION .. ==> NO      (NO or YES)
 5 DATA CURRENCY ..... ==> NO          (NO or YES)
 6 PARALLEL DEGREE ..... ==> ANY        (1 or ANY)
 7 RESOURCE ACQUISITION TIME ==> USE     (USE or ALLOCATE)
 8 REOPTIMIZE FOR INPUT VARS ==> NO      (ALWAYS, NONE, or ONCE)
 9 DEFER PREPARE ..... ==> NO          (NO or YES)
10 KEEP DYN SQL PAST COMMIT. ==> NO      (NO or YES)
11 DBPROTOCOL ..... ==>                (Blank, DRDA or PRIVATE)
12 APPLICATION ENCODING..... ==>        (Blank, ASCII, EBCDIC, UNICODE,
or ccsid)
13 OPTIMIZATION HINT ..... ==>          (Blank or 'hint-id')
14 IMMEDIATE WRITE..... ==> NO          (YES, NO)
15 DYNAMIC RULES ..... ==> BIND        (RUN or BIND)
16 SQLRULES ..... ==> DB2              (DB2 or STD)
17 DISCONNECT ..... ==> EXPLICIT       (EXPLICIT, AUTOMATIC,
F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND     F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE

```

Puc. 13

```
                                PACKAGE LIST FOR BIND PLAN                                SSID:
COMMAND ==>                                                                SCROLL ==>

Enter names to be included in PACKAGE list for: CBxxEz03

CMD      LOCATION      COLLECTION      PACKAGE-ID
"      "      "      "
"      "      TSOU$xx      pgmname
```

Puc. 14

Подрисуночные подписи

Рис. 1. Алгоритм подготовки программы

Рис. 2. Настройки утилиты DCLGEN

Рис. 3. Главное меню разрабатываемой программы

Рис. 4. Ввод имени работника

Рис. 5. Добавление нового работника в штат

Рис. 6. Меню подготовки программ DB2

Рис. 7. Меню настроек прекомпиляции программы

Рис. 8. Панель настроек связывания

Рис. 9. Настройки связывания по умолчанию

Рис. 10. Панель настроек подготовки программы к запуску

Рис. 11. Меню подготовки программы DB2

Рис. 12. Панель BIND PLAN

Рис. 13. Настройки BIND PLAN по умолчанию

Рис. 14. Выбор пакета для плана