

## Microstation и неграфические данные

Обычно частные предприятия и правительственные организации оперируют двумя наборами данных о предприятии или продукции:

- Архитектурные или инженерные чертежи проекта
- Неграфические данные, сохраняемые в базах данных организации

Один набор данных обычно не имеет самостоятельного применения без другого, однако часто наборы данных сохраняются по отдельности. Неграфические данные печатаются и добавляются к чертежам только после их вывода на плоттер. Хотя этот подход может быть удобен для производственного цеха, он может быть неэффективен для организатора проекта, которому необходимы информативные, полезные чертежи для реализации проекта.

Некоторые прикладные программы новой разработки связывают неграфические данные с чертежами — например, программы типа систем управления городским хозяйством и типа географических информационных систем (GIS) делают это очень хорошо. Общая особенность большинства таких систем - интерфейс базы данных, встроенный в приложение и скрытый от пользователя.

### Связывание графических элементов с неграфическими данными

MicroStation позволяет пользователю формировать приложение, которое связывает неграфические данные с графическими элементами (включая текст и фрагменты), используя стандартные интерфейсы MicroStation.

Эта связь может быть выполнена с использованием одного из двух средств MicroStation:

- Тэги элемента - в этом случае связываемые данные сохраняются в файле проекта вместе с графическими элементами. Использование тэгов целесообразно, если потребности в присоединении неграфических данных к элементам в файле проекта относительно просты или необходимо обеспечить совместимость с другими пакетами

Что такое - интерфейс базы данных?

CAD, которые хранят данные внутри файла чертежа. Дополнительную информацию, смотри в разделе "Тэги элементов" в главе 12 *Руководства пользователя*.

- **Интерфейс базы данных** - в этом случае подключаемые данные сохраняются в отдельной реляционной базе данных, которая связана с элементами в файле проекта.

Остальные главы этого руководства описывают интерфейс базы данных.

### Что такое - интерфейс базы данных?

Интерфейс базы данных - это набор инструментальных средств и установочных параметров, позволяющих связывать (или соединять) запись в реляционной базе данных с элементом в проекте. Это выполняется с сохранением уникального номера и в строке базы данных и в элементе в файле проекта.

Как только это соединение установлено, можно манипулировать и просматривать информацию базы данных изнутри MicroStation. Можно трансформировать любой запрос базы данных в графическое изображение или объединять запрос с выбранным набором или частью файла проекта, чтобы вывести на печать или плоттер отчеты, которые не возможно получить иным способом.

Непосредственно поддерживаются следующие коммерческие СУБД: Oracle, Xbase, INFORMIX, Intergraph's Relational Interface System (RIS) и Open Database Connectivity Standard (ODBC). Xbase - общее имя, используемое для описания семейства СУБД, поддерживающих формат файла dBASE III Plus. Смотри "Требования к памяти для сеансов MicroStation снижаются, если не подключена база данных (не загружен сервер)." на странице 11-6 для получения информации о платформах и доступных программах.

RIS и ODBC - не являются реляционными базами данных как таковые; - это обобщенный SQL интерфейс, который позволяет прикладным программам обращаться к базам данных, использующим SQL.

## **Почему следует использовать интерфейс базы данных?**

Использование интерфейса базы данных предлагает много возможностей, обычно не доступных пользователю CAD. Это диапазон возможностей от простого приложения, отслеживающего номера деталей в сборке, до сложной системы управления, которая отслеживает операции и сопровождает деятельность большой организации.

Например, графический элемент, представляющий улицу на карте, может быть связан с именем дороги или улицы, значением потока движения, числом несчастных случаев, датой строительства и информацией о ремонте в таблице базы данных. Центральная точка и текст номера лота каждого участка земли могли бы быть связаны с другой таблицей базы данных, включающей владельца, площадь, стоимость и имя улицы.

Связывая две таблицы, можно было бы, например:

- Выборочно отобразить все частные владения, которые имеют стоимость свыше 100, 000 долларов и интенсивность движения свыше 100, 000 транспортных средств в год, и подготовить отчет, включающий имя и адрес каждого владельца.
- Выборочно отобразить или подсветить все дороги в северо-восточном квадранте области, которые имеют интенсивность движения 50, 000 транспортных средств в день и свыше 10 несчастных случаев в год.
- Подготовить карту района, показывающую имена всех владельцев и стоимость частной собственности. При изменении базы данных можно глобально обновлять карту.

<& Использование интерфейса базы данных требует больше памяти, чем при стандартном использовании MicroStation.

Дополнительная память улучшает эффективность и MicroStation, и базы данных. Более подробную информацию о системных требованиях для использования MicroStation с какой-либо базой данных смотри в разделе "Требования к памяти" на странице 11-8.

## **Что такое - реляционная база данных?**

Реляционная база данных - это упорядоченная совокупность таблиц или файлов, представляющих объекты, их свойства и связи между объектами.

В этом руководстве используются следующие термины:

- **Реляционная база данных** представляет собой совокупность **таблиц**.
- Каждая таблица содержит **строки и столбцы**.

Обычно таблицы в базе данных имеют общий столбец или поле, который связывает их между собой. Отсюда термин реляционная база данных. В примере базы данных GIS, поставляемом с MicroStation, две таблицы: highways (скоростных дорог) и parcel (земельных участков) имеют общий столбец name (имя улицы).

*Таблица highways (скоростных дорог):*

mslink	name (имя улицы)	created (создана)	paved (проложена)	yearvolume (годовой поток)	speed limit (ограничение скорости)
2	BRIDGEWATER RD	1967	1983	176, 200	35
3	KILDARE OR	1978	1978	175, 200	35
6	KILLARNEY RD	1978	1978	176, 200	35

*Таблица parcel (земельных участков):*

mslink	parcel number (номер участка)	house number (номер дома)	name (имя улицы)	value (стоимость)	area (площадь)
1	9	220	BRIDGEWATER RD	9000	16, 036
14	36	8717	KILDARE DR	7200	16, 668
15	35	8713	KILDARE DR	9000	16, 306

Xbase не соответствует реляционной модели. База данных Xbase - эквивалент одной таблицы в реляционной модели. Следующий список показывает связь между терминами реляционной СУБД и терминами Xbase:

Реляционная	Xbase
Таблица	База данных (".dbf файл)
База данных	Совокупность файлов ".dbf
Строка	Запись
Столбец	Поле

Интерфейс базы данных Xbase в MicroStation имитирует модель реляционной базы данных, требуя, чтобы все связанные файлы базы данных Xbase находились в одном подкаталоге. Таким образом подкаталог становится реляционной базой данных.

## Почему используется реляционная база данных?

Имеется много причин для использования внешних реляционных баз данных, чтобы подключить текстовую информацию к графическим элементам. Реляционные базы данных - это наиболее популярные модели баз данных на рынке сегодня. Они разработаны для оперирования большими объемами данных. Обычно они используют язык Structured Query

Language (SQL) для доступа и модификации данных. Такие СУБД, как Oracle, INFORMIX и dBASE, наряду со стандартной моделью SQL, расширили возможности управления и манипулирования данными.

- ODBC Windows NT

Интерфейс фирмы Microsoft - Open Database Connectivity (ODBC) позволяет прикладным программам через драйверы базы данных осуществлять доступ к данным в ряде систем управления базами данных (СУБД). Это позволяет пользователям легко и быстро изменять среду базы данных, добавляя драйверы базы данных ODBC без изменений в ODBC совместимых приложениях.

## Сервер базы данных

Программа, реализующая интерфейс базы данных и MicroStation для конкретного пакета базы данных — например, Oracle — иногда называется **сервером базы данных**. **Модель сервера** определяет взаимодействие между MicroStation, пакетом базы данных и сервером базы данных. Если Вы не занимаетесь разработкой приложений баз данных для MicroStation, то нет необходимости глубоко знакомиться с описанием модели сервера. Достаточно знать, что в модели сервера функции базы данных отделены от графических функций. Это означает:

- Различные базы данных могут быть "подключены" через различные серверы.
- Серверы базы данных разработаны и поддерживаются отдельно от MicroStation.
- Требования к памяти для сеансов MicroStation снижаются, если не подключена база данных (не загружен сервер).

## Выбор базы данных

MicroStation поддерживает многие из популярных семейств реляционных баз данных, имеющихся на рынке сегодня. Выбор "наилучшей" СУБД затруднен, так как эти базы данных представляют широкий диапазон возможностей, эффективности и стоимости. Можно выполнять некоторые операции, как на простой однопользовательской базе данных семейства Xbase на PC, так и на полномасштабной корпоративной сетевой системе с серверами баз данных и различными базами данных, осуществляя при этом все взаимодействия с одними и теми же файлами проекта.

Набор доступных интерфейсов баз данных MicroStation изменяется в зависимости от используемой операционной системы следующим образом:

- Oracle — DOS, Windows 3.1, Windows 95, Windows NT, UNIX, Macintosh
- INFORMIX—UNIX
- Xbase — DOS, Windows 3.1x, Windows NT, UNIX, Macintosh
- RIS — DOS, Windows NT, UNIX (Рабочие станции Intergraph и SPARC/SunOS только)
- ODBC — Windows 3.1x, Windows 95, Windows NT

В настоящее время нет доступных баз данных для Windows 95. Однако, несколько баз данных, включая Xbase (DB4), Oracle и ODBC, будут скоро поддерживаться под Windows 95. Когда эти

базы данных станут доступными, MicroStation будет поддерживать их.

Эта глава описывает актуальные проблемы, которые необходимо учитывать при выборе базы данных, а также главные особенности различных поддерживаемых баз данных.

## **Факторы, определяющие выбор**

Имеется много факторов, влияющих на выбор базы данных. Например, если только один пользователь использует информацию базы данных, то можно было бы развернуть ее на системе пользователя. Обоснованным решением в этом случае могло бы быть приложение для семейства Xbase. Или, при необходимости в доступе к данным группы пользователей, можно выбрать установку СУБД на сетевом сервере.

Любая из поддерживаемых баз данных может выполняться на сетевом сервере, но при этом они работают совершенно по-разному. В случае Xbase, сервер только хранит файлы базы данных. Вся обработка выполняется на системе пользователя. Это может привести в результате к интенсивным обменам большими объемами данных на сети.

Oracle, INFORMIX и RIS - это базы данных типа клиент-сервер. В них ядро базы данных выполняется на сервере. Оно получает запросы от программ клиентов на каждой пользовательской системе. Запрос обрабатывается и только ответ передается по сети. Этот тип баз данных может поддерживать большое количество пользователей, благодаря снижению интенсивности обменов на сети. Эти СУБД могут также выполняться при совместной установке программ клиента и сервера на одной системе пользователя.

Можно выделить следующие критерии выбора, которые необходимо учитывать:

- Как будет установлена база данных - локально или на сетевом сервере?
- Будут ли обращения к данным из различных типов систем (например, смесь рабочих станций под управлением Windows NT и PC под DOS)?
- Будет ли СУБД использоваться независимо от MicroStation?
- Должно ли быть приложение MicroStation независимым от базы данных? То есть должно ли приложение выполняться без изменений с различными базами данных?
- Каков должен быть параметр Требования к памяти (см. страницу 11-8)?
- Будут ли требоваться только табличные отчеты с неграфическими данными?
- Будут ли корпоративные стандарты баз данных влиять на ваш выбор?
- Будет ли необходимо подключение к корпоративной базе данных?
- Будет ли необходимо обращаться к нескольким различным СУБД?

## **Требования к памяти**

Чтобы использовать интерфейс базы данных для Oracle или INFORMIX, система должна иметь достаточно памяти для одновременного выполнения MicroStation и базы данных. База данных сама добавляет по крайней мере 4 МБ к общей требуемой памяти. Дополнительная программа для экранных форм — SQL\* Forms (Oracle) или PERFORM (INFORMIX) — добавляет еще 1-2 МБ. Использование интерфейса Xbase не увеличивает требований к общей памяти.

## Семейства программ баз данных

Обычно СУБД это не одна программа или приложение, а семейство или набор прикладных инструментальных средств, позволяющих формировать, поддерживать и обрабатывать различную информацию. Поскольку MicroStation поддерживает широкий диапазон коммерческих баз данных, необходимо ознакомиться с возможностями каждого семейства прежде, чем сделать выбор.

### Oracle

Oracle, разработанная фирмой Oracle Corporation, является очень популярной базой данных, основанной на модели клиент-сервер.

Oracle предлагает несколько программ баз данных с различными возможностями, которые могут использоваться с MicroStation. Хотя между этими программами имеются значительные различия, MicroStation надо найти ответы на следующие вопросы:

- Требуется доступ к локальной или удаленной (на сервере) базе данных?
- Важно ли использование экранных форм (SQL\* Forms) внутри MicroStation?
- Потребуется ли доступ с нескольких различных типов систем?

В следующей таблице приведен список продуктов Oracle и требования для их использования с локальной и удаленной (на сервере) базой данных:

Продукт:	Локальная база данных	Удаленная база данных:
RDBMS	Обязательна	(Не используется)
SQL* Report Writer	Необязательна	Необязательна
SQL* Forms	Необязательна	Необязательна
SQL* Plus	Обязательна	Необязательна
SQL* Net	Необязательна*	Обязательна

\* Обязательный продукт для поддержки удаленных клиентов

#### Описания продуктов

Следующие продукты Oracle представляют интерес для пользователя MicroStation:

Продукт:	Описание
Oracle (базовый продукт)	Ядро реляционной СУБД. Оно обеспечивает возможности управления хранением данных, поиск и безопасность данных
Инструментальные средства разработки Oracle	

Oracle SQL* Forms	Инструментальные средства для разработки и управления формо-ориентированными приложениями в Oracle
Oracle SQL* ReportWriter	Непроцедурное инструментальное средство разработки приложений для генерации отчетов с выпадающими меню и мощными возможностями форматирования.
Oracle SQL* Menu	Инструментальное средство разработки меню четвертого поколения, которое обеспечивает общий внешний интерфейс к приложениям Oracle и других СУБД
Oracle Pro* C	Предоставляет разработчикам возможность включать операторы манипулирования базой данных в их приложения
Oracle SQL* Loader	Позволяет загружать таблицы баз данных из других источников в Oracle Это средство, используемое вместе с DB3PREP Oracle, упрощает преобразование файлов базы данных dBASE III Plus в формат Oracle. Эти инструментальные средства позволяют практически использовать Xbase, как платформу разработки прототипа для более сложных прикладных программ Oracle.
Инструментальные средства базы данных Oracle	
Oracle SQL* Plus	Обеспечивает интерактивный интерфейс Oracle с обеспечением специальных запросов и манипулирования базой данных
Инструментальные сетевые средства Oracle	
SQL* Net	Позволяет MicroStation обращаться к удаленным базам данных Oracle в сети.

## INFORMIX

INFORMIX, разработанная фирмой Informix Software, Inc., также основывается на модели клиент-сервер. В следующих таблицах приведены списки продуктов INFORMIX и требования для их использования с локальной и удаленной (на сервере) базой данных:

	Локальная база данных:	Удаленная база данных:
INFORMIX-SE		
SE Engine	Обязательна	(Не используется)
INFORMIX-SQL	Обязательна	Необязательна
ESQL Runtime	(Не используется)	Обязательна
INFORMIX-NET	Необязательна! <sup>а</sup>	Обязательна
INFORMIX-4GL	Необязательна! <sup>а</sup>	Необязательна! <sup>а</sup>
INFORMIX-OnLine		
Online Engine	Обязательна	(Не используется)
INFORMIX-SQL	Обязательна	Необязательна
ESQL Runtime	(Не используется)	Обязательна
INFORMIX-STAR	Необязательна <sup>б</sup>	Обязательна
INFORMIX-4GL	Необязательна	Необязательна

а. Необходима для поддержки удаленных клиентов.

б. Для экранных форм необходима (по крайней мере) исполняющая версия.

Описания продуктов

Следующие продукты INFORMIX представляют интерес для пользователя MicroStation:

Продукт:	Описание:
Базовые продукты базы данных INFORMIX	
INFORMIX-SE	Для компаний, в которых требуется обработка малых или средних объемов данных.
INFORMIX-OnLine	Обеспечивает решения для интерактивной диалоговой обработки запросов, включая графику, звук и текст.
Инструментальные средства базы данных INFORMIX	
INFORMIX-SQL	Обработчики запросов и средства проектирования настраиваемых экранных форм, отчетов и меню.
INFORMIX-4GL	Прикладные инструментальные средства разработки четвертого поколения для полноэкранных приложений, написания отчетов и ввода-вывода базы данных. В состав продукта включен 4GL интерпретатор и интерактивный отладчик.

Продукт:	Описание:
INFORMIX-ESQL	Включает инструментальные средства для разработки программ на языках третьего поколения типа Си и ФОРТРАН. INFORMIX -ESQL содержит встроенный SQL, который позволяет этим программам обращаться к любой базе данных INFORMIX.
Инструментальные средства связи INFORMIX	
INFORMIX-Net	Поддерживает распределенную обработку прикладных инструментальных средств и базовых модулей базы данных.
INFORMIX-STAR	Обеспечивает использование многих баз данных, установленных на различных системах в сети. Базы данных могут быть связаны и просмотрены таким образом, как будто бы они были установлены на одной системе.

## Xbase

Xbase - это термин, используемый для описания форматов файла данных/индексного файла ряда СУБД, которые базируются на популярном формате dBASE III Plus. В смысле способа хранения данных единственное, что отличает эти продукты - это формат индексного файла.

Поддерживаются следующие форматы индексов Xbase:

- dBASE IV (.mdx)
- dBASE III Plus (.ndx)
- FoxPro (.cdx)

Продукты Xbase содержат полный набор инструментальных средств для создания и поддержки баз данных, включая мощный язык команд dBASE и язык четвертого поколения (4GL) для разработки генераторов отчетов. Кроме того несколько третьих фирм предлагают непроцедурные генераторы отчетов, использующие формат файла dBASE III Plus.

Интерфейс Xbase MicroStation отличается от интерфейсов для Oracle, INFORMIX и RIS тем, что он не обращается к базе данных, а читает и пишет данные в базу данных и индексные файлы непосредственно. Таким образом на уровне пользователя (или приложения) нет необходимости в других продуктах. Подобно другим интерфейсам, интерфейс Xbase поддерживает SQL. С



точки зрения пользователя, интерфейс Xbase функционирует также как другие интерфейсы базы данных.

## **Интерфейс базы данных ODBC для Windows NT**

Интерфейс фирмы Microsoft Open Database Connectivity (ODBC) обеспечивает, через драйверы баз данных, приложениям доступ к данным в ряде СУБД. Это позволяет пользователям легко и быстро изменять среду базы данных, добавляя драйверы базы данных ODBC без изменений в ODBC совместимых приложениях.

Чтобы использовать интерфейс ODBC, необходимо следующее:

- Windows NT 3.5, Windows 3.1x
- Intersolv Data Direct (прежде Q+ E) ODBC NT драйверы версии 1.2 или позже для следующих баз данных: dBASE IV, Oracle 6 и 7, SQL Server 4.2x, Microsoft Access 2.0.

<& Настоящий продукт не содержит драйверы ODBC для

поддерживаемых баз данных. Пользователь должен приобрести драйвер (ы) ODBC для сервера ODBC MicroStation. MicroStation поддерживает только Intersolv ODBC NT драйверы. Данная версия поддерживает только те конфигурации, которые удовлетворяют вышеупомянутым требованиям. Никакие другие базы данных, кроме вышеупомянутых, не были сертифицированы для этой версии MicroStation.

## **RIS**

RIS (Relational Interface System) - продукт фирмы Intergraph. Он позволяет приложению обращаться к ряду различных СУБД (включая Oracle, INFORMIX, Ingres, Sybase, Rdb, DB2 и IBM AS/400) через общий интерфейс. Приложение связывается с RIS, а RIS обеспечивает необходимую трансляцию в протокол, понятный основному модулю базы данных. Это дает возможность переключать приложения, написанные для RIS, между различными коммерческими базами данных без изменения их исходного текста. Интерфейс пользователя RIS независим от присоединенной базы данных.

Интерфейс RIS в основном состоит из трех частей: приложение (RIS интерфейс MicroStation), клиент и сервер данных. Приложение поставляется вместе с MicroStation. RIS клиент и сервер данных должны быть приобретены отдельно у Intergraph. Клиент обычно выполняется на той же самой системе, что и приложение. Сервер данных выполняется на той системе, где находится ядро базы данных. Это может быть общая система, на которой выполняется программное обеспечение клиента и приложения, или удаленная главная ЭВМ, расположенная где-нибудь на сети. Исключение составляет DOS RIS. Под DOS

программное обеспечение клиента не располагается на той же самой системе, что и приложение.

### **Инструментальные средства RIS**

RIS имеет полный набор инструментальных средств для создания и управления базами данных, формулирования запросов, проектирования настраиваемых экранных форм и генерации отчетов.

RIS Client и Runtime Package необходимы для выполнения RIS приложений.

RIS Development Platform необходим только разработчикам RIS приложений.

RIS Utilities (необязательны) необходимы для выполнения задач типа схематичное управление, специальные запросы и загрузка больших объемов данных.

RIS Report Writer обеспечивает генерацию отчетов и редактирование реляционных баз данных. Он автоматизирует, стандартизирует и упрощает сетевое взаимодействие. Созданный на базе I/FORMS, стандартного графического интерфейса фирмы Intergraph, RIS Report Writer позволяет выбирать функции и значения из меню для генерации команд SQL. Любой пользователь может быстро обучиться генерации отчетов и редактированию баз данных.

Прежде чем использовать интерфейс базы данных MicroStation, необходимо выполнить процедуры установки. Эти процедуры описываются в этой главе.

## Общая процедура установки интерфейса базы данных

### 1. Установка базы данных для MicroStation:

База данных:
Интерфейс ODBC для Windows NT (страница 12-2)
Oracle (страница 12-5)
INFORMIX (страница 12-12)
Xbase (страница 12-12)
RIS (страница 12-17)

### 2. Установка MicroStation для использования базы данных:

База данных:
Oracle (страница 12-17)
INFORMIX (страница 12-19)
Xbase (страница 12-20)
MS (страница 12-23)

Некоторые дополнительные установки необходимо проделать прежде, чем выполнять уроки в следующих нескольких главах для изучения интерфейса баз данных. Более подробную информацию смотри в разделе "Установка" в главе 14 *Учебного пособия*.

## Установка базы данных для MicroStation

В следующих разделах описано, как установить ODBC, Oracle, INFORMIX, Xbase или базу данных RIS для использования с MicroStation.

Символ прямого слэша "/" используется для отделения каталогов в спецификациях пути в процедурах установки. Фактический символ разделителя может быть другим, в зависимости от используемой операционной системы.

## Интерфейс ODBC для Windows NT

Интерфейс фирмы Microsoft - Open Database Connectivity (ODBC) позволяет прикладным программам через драйверы базы данных осуществлять доступ к данным в ряде систем управления базами данных (СУБД). Это позволяет пользователям легко и быстро изменять среду базы данных, добавляя драйверы базы данных ODBC. Без внесения изменений в ODBC совместимые приложения.

Чтобы использовать интерфейс ODBC, необходимо следующее:

- Windows NT 3.5, Windows 3.1
- Intersolv Data Direct (прежде Q+ E) ODBC NT драйверы версии 1.2 или более поздние для следующих баз данных: Fox Pro, dBASE III Plus, dBASE IV, Clipper, Oracle 6 и 7, SQL Server 4.2x и INFORMIX.
- Драйвер Microsoft ODBC Desktop Driver Pack 2.0 Access driver

Настоящий продукт не содержит драйверы ODBC для поддерживаемых баз данных. Пользователь должен приобрести драйвер (ы) ODBC для сервера ODBC MicroStation. Для получения дополнительной информации относительно InterSolv ODBC Direct Drivers можно позвонить по телефону 1-800-876-3101. Данная версия поддерживает только конфигурации, удовлетворяющие вышеупомянутым требованиям. Для данной версии MicroStation никакие другие базы данных, кроме вышеупомянутых, не были сертифицированы.

## Установка интерфейса ODBC

Подобно RIS, ODBC использует источник данных для хранения информации о соединении с базой данных. MicroStation использует имя источника данных ODBC для команды соединения с базой данных — `OB=<имя_источника_данных>`. Драйвер (ы) ODBC должен быть загружен до создания источника данных. Как только драйверы загружены, в Control Panel (панели управления) Windows появляется пиктограмма ODBC.

1. Открыть администратор драйвера ODBC - диалоговое окно Data Sources, двойным нажатием на пиктограмме ODBC в Control Panel (панели управления).
2. Нажать кнопку Add.
3. Выбрать драйвер ODBC для СУБД, содержащий требуемые данные.

Появляется диалоговое окно источника данных.

4. Определить имя источника данных (это строка, используемая в команде MicroStation `"DB="`) и другую информацию, относящуюся к используемой конфигурации.

Поля Xbase и Oracle требуют специфических установочных параметров, которые описаны ниже. Следует обратиться к руководству драйвера ODBC для изучения специфических особенностей других полей.

Xbase

"Locking" является уровнем блокирования файла dbf. Поле блокирования должно быть установлено в значение "File". Это гарантирует многим пользователям отсутствие конфликтов активных объектов. Если индексные файлы имеют то же имя как файлы dbf, то не обязательно щелкать кнопку Define, чтобы связать индексный файл с файлом dbf.

Oracle

'Server Name' является строкой удаленного доступа Oracle. Если необходимо соединиться дистанционно с сервером Oracle, то надо ввести строку удаленного доступа Oracle, то есть T: MY\_SERVER: DEMO. Если используется локальная база данных, то надо оставить это поле пустым.

DSN (Data Source Name) должен быть определен в ODBC Data Source. Идентификаторы ID сервера и пользователя могут быть также определены в ODBC Data Source; а пароль - нет. Пароль может однако быть установлен, как переменная MS\_ODBCPARAMS. Можно также установить идентификаторы ID сервера и пользователя, как переменные MS\_ODBCPARAMS.

Проверьте последнюю строку в файле <MSDIR> database/ config/ ODBC.CFG она должна иметь вид MS\_ODBCPARAMS=DLG=2. Если ввести с клавиатуры неполную или неправильную

информацию для связи с Oracle, то будет возвращено сообщение об ошибке "Оператор connect не выполнен".

### **Доступ ко многим СУБД**

Теоретически стандарт ODBC позволяет ODBC совместимому приложению иметь доступ к данным из любой СУБД, имеющей драйвер ODBC. Практически имеются несколько моментов, когда возникают проблемы при доступе ко многим СУБД.

Стандарт ODBC основан на SQL базах данных. Все базы данных с плоскими файлами (Btrieve; dBASEII, III и IV; Clipper; FoxPro;

Excel; и Paradox) должны иметь уровень SQL, который транслирует вызов SQL ODBC в язык, понимаемый СУБД. Проблема в том - что разработчик драйвера ODBC должен реализовать уровень SQL интерпретатора. Синтаксис SQL в этом случае зависит от того, как полно разработчик придерживается ANSI и промышленных стандартов.

### **Общие примечания**

Ниже приведена некоторая общая информация об ODBC:

- DBCHECK не будет работать со связями ODBC.
- Информация об элементе (analyze.ma) не будет отображать связи ODBC. Они входят в список User Data Linkages.

### **Примечания к DBASE**

В MicroStation версии 5.0 интерфейс Xbase использовал словарь данных, чтобы перейти к парадигме SQL. Сервер ODBC не использует словарь данных, так что пользователь должен знать некоторые ограничения dBASE:

Имена файлов не могут содержать более восьми символов. MSCATALOG - это имя mscatalog для интерфейса Xbase ODBC. Если вызвать таблицу mscatalog, используя MSCATALOG как имя таблицы, то будет выдано сообщение об ошибке ODBC 4101:

"соединение, предложение или обработка запроса недействительны. "

Имена столбцов не могут содержать более 10 символов. Такое же ограничение было в интерфейсе MicroStation версии 5. Однако в интерфейсе ODBC, вместо усечения имен столбцов, превышающих 10 символов, библиотечные функции возвращают ошибку.

Индексные файлы не могут содержать выражения. Индексные файлы примера Xbase GIS, поставляемого с MicroStation, (и возможно некоторые из ваших индексных файлов) содержат выражение "FOR NOT DELETED O". Intersolv драйвер Xbase ODBC не может обращаться к информации в таблице, имеющей

индексный файл, который содержит выражение, выражения должны быть удалены Эта ошибка должна быть устранена в будущих реализациях драйвера ODBC от Intersolv

#### Примечания к SQL Server

SQL Server не распознает типы данных SQL ANSI Ниже приводится перечень распознаваемых типов money, small money, int, smallint, tinyint, char (n), varchar (n), text, datetime, small datetime, bit, timestamp, sysname, float, real, binary (n), varbinary (n), image

и SQL Server не распознает следующие типы данных number, numeric, decimal, integer, and logical

SQL Server обрабатывает NULL способом, несовпадающим с ANSI В то время как стандарты ANSI определяют, что столбец по умолчанию может принимать значения NULL, в SQL Server необходимо определить столбец как NULL, чтобы он мог принимать значения NULL. Опция NOT NULL на столбцах работает так, как ожидается Это может вызывать проблемы при создании активного объекта базы данных Если столбец в таблице активного объекта не имеет значения, то активный объект не может быть вставлен в таблицу, если предложение, создающее таблицу, не определило пустой столбец с помощью опции NULL

В настоящее время Intersolv драйвер SQL Server ODBC не поддерживает утверждения SQL свыше 2400 символов

Данный раздел содержит описание установки базы данных Oracle для использования с MicroStation. Для получения общей информации о семействе продуктов Oracle смотри "Oracle" на странице 119

## Общая процедура установки Oracle

1. Выполнить специальные шаги установки из раздела "примечания" для вашей операционной системы.

Система:	Смотри:
DOS	Примечания к DOS (страница 12-7)
Windows NT	Примечания к Windows NT 3.5 (страница 12-6)
UNIX	Примечания к UNIX (страница 12-11)
Macintosh	Примечания к Macintosh (страница 12-11)

2. Запустить Oracle. Этот шаг должен выполняться перед любым сеансом MicroStation, в течение которого происходит соединение с базой данных Oracle. Смотри раздел "примечания" для вашей операционной системы. (Администратор базы данных обычно ответственен за правильный запуск Oracle.)

3. В SQL\* Plus выполнить имеющийся командный файл SQL - "mscat.sql" для создания таблицы, описанной в разделе Таблица MSCATALOG (см. страницу 13-7) и других управляющих таблиц, вызываемых изнутри MSCATALOG. Смотри раздел "примечания" для вашей операционной системы.

| Если используется программа экранных форм SQL\* Forms с интерфейсом базы данных Oracle, то необходимо создать соответствующие триггеры для загрузки экранных форм. Дополнительную информацию относительно создания триггеров

для загрузки смотри в разделе "Установка процедур загрузки экранных форм" на странице 12-27.

### **Примечания к Windows NT 3.5**

При установке Oracle никаких специальных шагов выполнять не требуются для использования с MicroStation под Windows NT. Обратитесь к Главе 4 в *Database Guide* MicroStation версии 5.0. Однако необходимо запустить Oracle, соединиться с базой данных и создать таблицу MSCATALOG.

Нет необходимости в теплом запуске Oracle для Windows NT, так как это будет выполнено автоматически при двойном нажатии на Instance Manager и запуске Oracle Instance.

#### **Чтобы выполнить соединение с базой данных Oracle из MicroStation под Windows NT:**

- В командной строке MicroStation ввести с клавиатуры  
OB=<имя\_пользователя>/<паспорт>@ (<адрес>)  
ИЛИ

1. Из подменю База данных меню Установки выбрать пункт Соединить.

Откроется диалоговое окно Соединение с базой данных.

2. Из меню опций Сервер БД выбрать Oracle.

3. В поле Строка связи ввести с клавиатуры

<имя\_пользователя>/<паспорт>@ (<адрес>)

4. Щелкнуть кнопку ОК.

Если MSCATALOG не существует, то окно предупреждения сообщает, что необходимо создать MSCATALOG для создания связей от элементов в проекте к строкам в базе данных.

#### **Чтобы создать таблицу MSCATALOG:**

1. Использовать SQL Plus, DBA или графическое средство SQL Oracle.

2. Выполнить командный файл "mscat.sql". В приглашении SQL> ввести:

@\win32app\ustation\database\oracle\mscat.sql

(Этот шаг предполагает, что каталог MicroStation -  
"\win32app\ustation.")

3. Выход.

ИЛИ

1. Из подменю База данных меню Установки (в MicroStation) выбрать пункт Настройка.

Откроется окно предупреждения с вопросом о необходимости создания MSCATALOG.

2. Щелкнуть кнопку Да.

MSCATALOG создается и открывается установочное окно MSCATALOG.

### **Примечания к DOS**

При установке Oracle для использования с MicroStation PC требуется выполнить специальные шаги .

#### *Использование памяти*

Как MicroStation так и Oracle используют расширение возможностей PC для адресации памяти вне ограничения стандартных 640 КБ. Это выполняется с помощью программы-расширителя DOS, которая обеспечивает доступ ко всей физической памяти, установленной в системе.

По умолчанию, Oracle использует всю расширенную память в системе. При теплом запуске базы данных вся расширенная память перераспределяется для использования Oracle и не доступна MicroStation.

Следовательно необходимо ограничить объем памяти для Oracle, установив параметр DYNAMIC\_MEMORY в "config.ora". Этот параметр определяет объем памяти в Кб, доступный базе данных Oracle и ее инструментальным средствам.

#### **Чтобы установить параметр динамической памяти Oracle:**

1. Используя текстовый редактор, добавить следующую строку к "config.ora":

```
DYNAMIC_MEMORY=<KB>
```

Например, если система имеет 16 МБ памяти и необходимо разделить ее поровну между Oracle и MicroStation, следует добавить следующую строку к "config.ora":

```
DYNAMIC_MEMORY=8192
```

#### *Управление памятью*

При конфигурировании Oracle необходимо определить тип используемой ЭВМ. Это определяет тип управления памятью, используемый Oracle. Возможные типы ЭВМ делятся на две категории: VCPI и не-VCPI. Если выбрать VCPI, то нет необходимости выполнять специальный шаг установки для MicroStation.

Если выбрать другой тип ЭВМ, то необходимо установить системную переменную DOSX с переключателем EXT LOW, чтобы сообщить MicroStation, где в памяти может начинаться ее загрузка.

Чтобы установить DOSX с переключателем EXT LOW:

1. В системной подсказке ввести:

```
SET DOSX - -EXTLOW <value>
```

Value- шестнадцатиричное число, вычисленное прибавлением 10000 lh к значению параметра DYNAMIC\_MEMORY в "config.ora" (смотри "Использование памяти" на странице 12-8). Например, если DYNAMIC\_MEMORY установлен в 8192 (Кб), которое равно 80000h (шестнадцатиричное), соответствующее значение EXTLOW value было бы 90001h. Можно добавить эту строку к "ustation.bat."

#### *Номера программных прерываний*

Номера программных прерываний, используемые Oracle, должны быть изменены так, чтобы они не конфликтовали с используемыми MicroStation.

#### **Чтобы изменить номера программных прерываний Oracle:**

1. Используя текстовый редактор, добавить следующие строки к "config.ora:"

```
INTERRUPT=102
```

```
XMMITR=240
```

Вывод SQL\* Forms

Имеется опция вывода SQL\* Forms в специальном окне DOS внутри MicroStation.

#### **Чтобы установить вывод SQL\* Forms Oracle в окне DOS:**

1. Используя текстовый редактор, добавить следующую строку к "config.ora:"

```
ORACLE_TIO=ANSI
```

Этот оператор установит режим отображения Oracle в ANSI. Хотя это не самый быстрый режим отображения, он будет наиболее совместим с аппаратными средствами. Окно DOS работает только с режимом отображения ANSI.

2. Используя текстовый редактор, добавить строку к файлу "config.sys" в корневом каталоге, чтобы загрузить драйвер "ansi.sys". Например:

DEVICE-c: \dos\ansi.sys

3. В системной командной строке ввести:

copy \oracle6\dfs\ansi.crt \oracle6\dfs\default.crt

**Этот оператор заставит Oracle читать файл "default.crt", чтобы получить информацию о видео атрибутах дисплея.**

**Чтобы выполнить теплый запуск Oracle в DOS:**

Имеются два метода: локальный и удаленный.

#### **Локальный**

1. Войти в Oracle.

2. В системной командной строке ввести:

oracle

При локальной конфигурации предложение соединения имеет следующий вид (имя\_пользователя/паспорт):

user\_name/password

#### **Удаленный**

1. В системной командной строке ввести:

sqlpme

Это запускает драйвер памяти защищенного режима Oracle.

2. В системной командной строке ввести:

<Oracle SQL\* NET executable>

Или для SQL\* Net TCP/IP ввести:

<Oracle sqltsp>

Загрузка SQLPME позволяет инструментальным средствам Oracle SQL\* Net использовать память защищенного режима вместо реального режима. Необходимо использовать SQLPME, если не имеется достаточно реальной памяти, чтобы загрузить Oracle и MicroStation одновременно.

При удаленной конфигурации все предложения соединения Oracle выглядят следующим образом:

useri\_name/password@NETWORK\_ID: MACHINE\_NAME: ORACLE\_SID

**Чтобы создать таблицу MSCATALOG из SQL\* Plus:**

1. Войти в SQL\* Plus. В системной командной строке ввести:

sqlpl us

2. Выполнить командный файл "mcat.sql". В приглашении SQL> ввести оператор:

@database/oracle/mcat.sql

3. Выйти из SQL\* Plus.

#### **Примечания к UNIX**

При установке Oracle для использования с MicroStation под UNIX никаких специальных шагов выполнять не требуются. Однако, их необходимо выполнить для теплового запуска Oracle и создания таблицы MSCATALOG из SQL\* Plus.

**Чтобы выполнить теплый запуск Oracle:**



1. Войти в Oracle.
2. В системной командной строке ввести:  
sqldb
3. В приглашении SQL\* DBA ввести:  
startup

**Чтобы создать таблицу MSCATALOG из SQL\* Plus:**

1. Войти в SQL\* Plus. В командной строке ввести:  
SQL plus
2. Выполнить командный файл "mcat.sql". В приглашении SQL> ввести:  
@database/oracle/mcat.sql
3. Выйти из SQL\* Plus.

**Примечания к Macintosh**

При установке Oracle для использования с MicroStation на системе Macintosh никакие специальные шаги выполнять не требуются. Однако, их надо выполнить для теплого запуска Oracle и создания таблицы MSCATALOG из SQL\* Plus.

**Чтобы выполнить теплый запуск Oracle:**

1. Запустить приложение Oracle для Macintosh из папки "Applications" Oracle.
2. Из меню Database выбрать пункт Startup Oracle.

**Чтобы создать таблицу MSCATALOG из SQL\* Plus:**

1. Запустить приложение SQL\* Plus из папки "Applications" Oracle.
2. Войти в Oracle.
3. Выполнить командный файл "mcat.sql". В приглашении SQL > ввести:  
@HD: MicroStation95: database: Oracle: mcat.sql  
(Этот шаг предполагает, что папка MicroStation называется "MicroStation 95" и она расположена на томе "HD.")
4. Выйти из SQL\* Plus.

## INFORMIX

В этом разделе описывается, как установить базу данных INFORMIX для использования с MicroStation. Общую информацию о продуктах семейства INFORMIX смотри в разделе "INFORMIX" на странице 11-10.

**Чтобы установить INFORMIX:**

1. Выполнить командный файл SQL - "mcat.sql", поставляемый с MicroStation, для создания таблицы, описанной в разделе Таблица MSCATALOG (см. страницу 13-7) и другие управляющие таблицы, вызываемые изнутри MSCATALOG. В системной командной строке ввести:

isql - mcat.sql

Если использовать программное обеспечение экранных форм PERFORM с интерфейсом базы данных INFORMIX, то необходимо выполнить следующее:

- Создать соответствующие 4GL программы для загрузки экранных форм. Для получения более подробной информации смотри "Установка процедур загрузки экранных форм" на странице 12-27.
- Установить системную переменную INFORMIX DBPATH, указывающую на каталог, который содержит ".per" файлы PERFORM.

## Xbase

В этом разделе описывается, как установить базу данных Xbase для использования с MicroStation. Для получения общей информации о продуктах Xbase смотри "Xbase" на странице 11-12.

Словарь данных для XBASE обучающих примеров предварительно был сформирован. Смотри Общая процедура "настройки для упражнений с Xbase" на странице 12-32.

Типовой командный файл "build.s" поставляется вместе с примерами MicroStation 95 XBASE. Чтобы загрузить этот файл, необходимо вначале удалить файлы, созданные при выполнении утилиты datadict. Выполнить командный файл "removedd.bat", поставляемый вместе с примерами, для удаления этих файлов и файла MSFORMS.DBF, который был создан при выполнении MicroStation.

Если Ваш словарь данных находится не в том же каталоге, где файлы базы данных и индексные файлы, то необходимо описать путь в командной строке. Например, командная строка для поставляемых примеров XBASE:

```
..\..\datadict -sbuild.s -obuild.rpt
```

**Файл build.rpt создается при выполнении утилиты datadict.**

### Словарь данных

Поскольку каждая таблица Xbase сохраняется в отдельном файле, а индексные файлы Xbase могут содержать множество индексов, то программное обеспечение интерфейса базы данных Xbase поддерживает словарь данных. Словарь данных связывает имена таблиц с фактическими именами файлов и включает индексные файлы, связанные с таблицами. Имена таблиц могут содержать до 10 символов. Имя файла ограничивается операционной системой (например, в случае DOS - 8 символов). Чтобы MicroStation могла обратиться к таблице, в словаре данных должна быть строка входа в соответствующую таблицу.

### Утилита Datadict

Утилита Datadict, поставляемая в составе MicroStation, используется для добавления к словарю данных таблиц и индексов, созданных внешними приложениями. Имеется версия утилиты для каждого из следующих форматов индексного файла базы данных:

Формат индексного файла	Версия Datadict
dBASE IV (.mdx)	"\$ (MSDIR) database/xbase/datadict.exe"
dBASE III Plus (.ndx)	"\$ (MSDIR) database/xbasendx/datadndx.exe"
FoxBase (.cdx)	"\$ (MSDIR) database/xbasefox/datadfox.exe"

Можно скопировать соответствующую версию в каталог, определенный в текущем пути.

Утилита Datadict всегда оперирует со словарем данных в текущем каталоге. При этом нет необходимости иметь добавляемые файлы и индексы в том же самом каталоге, где и словарь данных, но это рекомендуется. Файлы базы данных, созданные MicroStation (или MDL приложениями), размещаются в каталоге базы данных

### **Режим командного файла Datadict**

Три файла - "datadict", "datadndx" и "datadfox" реализуют режим командного файла для программ утилиты Datadict. Режим командного файла позволяет пользователям определять командный файл, содержащий все аргументы Datadict и командный файл, содержащий предупреждающие сообщения. Поставляемый командный файл, содержащий предупреждающие сообщения, пуст

Синтаксис: datadict -зима\_командного\_файла  
-о имя\_выходного\_файла

Командный файл может выглядеть следующим образом

init

```
ta parcel parcel
ia parcel parcel
ta pard parcel_das
ta maps maps
ia maps maps
ta features features
ia features features
ta highway highway
ia highway highway
ta mscatlog mscatlog
```

Командный файл подобен поставляемому файлу build.bat для примера gis, за исключением того, что имя утилиты datadict отсутствует во всех строках

<& Внешние приложения (т.е. не MicroStation MDL приложения), которые обращаются к таблицам, используемым программным обеспечением интерфейса баз данных Xbase, должны очень осторожно изменять структуры базы данных. Например, если внешнее приложение добавляет некоторый столбец или некоторые строки к таблице, но не изменяет соответствующий индексный файл, то словарь данных теряет синхронизацию. Для восстановления синхронизации необходимо вновь использовать

утилиту Datadict или же внешнее приложение должно обеспечивать изменение словаря данных вручную. (Последнее решение может быть проблематичным, если MicroStation (или MDL приложение) обращаются к базе данных в то же самое время, как и внешнее приложение.)

Режим командного файла используется для Win32s и всех других платформ, исключая Macintosh.

### **Чтобы установить Xbase:**

1. (Необязательно) Создать управляющую таблицу MSCATALOG, используя программное обеспечение базы данных. Структура таблицы описана в разделе "Таблица MSCATALOG" на странице 13-7. Вместо создания таблицы с самого начала, можно скопировать и изменить пример, представленный в "mscatlog.dbf". Этот файл MicroStation записан в каталоге "examples/gis" для используемого формата индексного файла.

Если этот шаг (и шаг 3) пропущены, то необходимо создать MSCATALOG после соединения с базой данных из MicroStation. Это можно выполнить, введя предложение SQL CREATE TABLE в окне SQL MicroStation.

2. Инициализировать словарь данных. В системной командной строке ввести:

```
datadict [ datadndx | datadfox> -init
```

3. (Пропустить, если не выполнен шаг 1). Добавить строку в словарь данных для управляющей таблицы MSCATALOG. В системной командной строке ввести:

```
<datadict | datadndx | datadfox> -ta mscatalog  
mscatalog
```

4. Добавить строку к словарю данных для первой таблицы данных. В системной командной строке ввести следующую команду:

```
<datadict | datadndx | datadfox> -ta <.dbf_file_name>  
<table_name>
```

Если table\_name опущено, то предполагается, что имя таблицы совпадает с именем базового файла с расширением ".dbf". Если словарь данных уже содержит строку для таблицы с этим именем, то команда не выполняется. Имя таблицы table\_name может содержать до 10 символов.

5. Добавить строку к словарю данных для индексного файла, связанного с таблицей, определенной на шаге 4. В системной командной строке, ввести следующую команду:

```
<datadict | datadndx | datadfox> -ia  
<index_file_name> <table_name>
```

Утилита не может открыть индексный файл по умолчанию с тем же самым именем. Если другой индексный файл связан с таблицей, то команда не выполняется. Текущий индексный файл должен быть вначале отключен. Смотри раздел "Чтобы отключить таблицу от ее индексного файла:" на странице 12-16.

6. Повторить шаги 4-5 для каждой таблицы, в которую необходимо добавить строку к словарю данных.

7. (Только DOS) Загрузить утилиту DOS SHARE в память так, как описано в разделе "Примечания к DOS" на странице 12-16.

Чтобы упростить выполнение вышеупомянутой процедуры, можно записать соответствующие команды Datadict в командный файл или файл сценария и при необходимости выполнять этот файл. Пример такого файла - "build.bat". Он находится в каталоге "examples/gis" для используемого формата индексного файла.

#### **Чтобы отключить таблицу от ее индексного файла:**

1. В системной командной строке ввести следующую команду:

```
<datadict | datadndx | datadfox> -id  
<index_file_name> <table_name>
```

Индексный файл не удаляется. Если индексный файл не связан с этой таблицей в словаре данных, то команда игнорируется.

#### **Чтобы удалить строку таблицы из словаря данных:**

1. В системной командной строке ввести следующую команду:

```
<datadict | datadndx | datadfox> -td <table_name>
```

Сам файл не удаляется.

#### **Примечания к DOS**

Утилита DOS, называемая SHARE, должна быть загружена в память до активизации интерфейса баз данных Xbase с MicroStation PC .

Чтобы загрузить утилиту SHARE:

1. В системной командной строке ввести следующую команду, указав каталог "dos" в описании пути:

share

Эта командная строка обычно включается в "autoexec.bat" вместо ее выполнения в системной командной строке.

## RIS

В этом разделе описывается установка RIS для использования с MicroStation.

### Общая процедура установки RIS

1. Создать схему RIS. Смотри документацию по RIS.

**2. Создать управляющую таблицу MSCATA LOG и управляющие таблицы, вызываемые из MSCATALOG:**

Используя RIS утилиту Load, выполнить файл "gis.dmp", поставляемый с MicroStation. Смотри документацию по RIS. Эта утилита создает типовые таблицы для упражнений из этого руководства, в дополнение к MSCATALOG (смотри "Установка процедур загрузки экранных форм" на странице 12-27).

или

Выполнить процедуру установки основной базы данных (смотри "Oracle" на странице 12-5 или "INFORMIX" на странице 12-12).

## Установка MicroStation для соединения с базой данных

Установка MicroStation для соединения с базой данных - это в значительной степени вопрос определения переменных конфигурации. Большинство этих определений указывает на каталоги и MDL приложения, относящиеся к операциям с интерфейсом баз данных. Одна из них - MS\_LINKTYPE (см. страницу 12-25) - рассматривается в отдельном разделе.

## Oracle

Следующие переменные конфигурации должны быть определены до соединения базы данных Oracle из MicroStation. Большинство

этих переменных конфигурации находятся в категории База данных в диалоговом окне Конфигурация. \$ (MSDIR) - сокращение для корневого каталога MicroStation. Дополнительную информацию относительно определения переменных конфигурации смотри в разделе "Работа с переменными конфигурации" в главе 4 *Руководства пользователя*.

Назначение	Имя переменной	Определение
Файлы базы данных	MS_DBASE	\$ (MSDIR) database/oracle/
Загрузчик сервера	MS_SERVER	dbload

Сервер базы данных	MS_DBEXT	orserver.pxe (DOS) orserver.exe (Windows NT) orserver (UNIX) orserver (Macintosh)
Связи базы данных	MS_LINKTYPE	ORACLE (смотри "MSLINKTYPE" на странице 12-25)
Загружаемые приложения проекта (категория Приложения проекта диалогового окна Конфигурация)	MS_DGNAPPS	Сервер
N/A (необязательно)	MS_USERMSC	Имя таблицы замещения для MSCATALOG. Это руководство всегда ссылается на таблицу как на MSCATALOG.

Чтобы подключить базу данных Oracle из MicroStation:

1. Ввести с клавиатуры:

ACTIVE DATABASE <имя\_пользователя>/<паспорт>

или

ОВ=<имя\_пользователя>/<паспорт>

Или использовать диалоговое окно Соединение с базой

данных, вызываемое из меню Установки>База

данных> Соединить

Возможно также подключить Oracle, вводя с клавиатуры |connect <имя\_пользователя>/<паспорт> или введя в окне SQL "connect <имя\_пользователя>/<паспорт>".

Примечания к DOS

Программа конфигурации MicroStation PC (для DOS) автоматически устанавливает требуемые определения переменных конфигурации, если выбрать пункт Oracle из меню опций Сервер БД. MS\_LINKTYPE (см. страницу 12-25) устанавливается в состояние записи связей с ORACLE и чтения связей всех других типов.

## INFORMIX

Следующие переменные конфигурации должны быть определены до соединения с базой данных INFORMIX из MicroStation. Большинство этих переменных конфигурации находятся в категории База данных в диалоговом окне Конфигурация. \$ (MSDIR) - сокращение для корневого каталога MicroStation.

Дополнительную информацию относительно определения переменных конфигурации смотри в разделе "Работа с переменными конфигурации" в главе 4 *Руководства пользователя*.

Назначение	Имя переменной	Определение
Файлы базы данных	MS_DBASE	\$ (MSDIR) database/ Informix/
Загрузчик сервера	MS_SERVER	dbload
Сервер базы данных	MS_DBEXT	ixserver (UNIX)

Связи базы данных	MS_LINKTYPE	INFORMIX (смотри "MS_LINKTYPE" на странице 12-25)
Загружаемые приложения проекта (категория Приложения проекта в диалоговом окне Конфигурация)	MS_DGNAPPS	Сервер
N/A (необязательно)	MS_USERMSC	Имя таблицы замещения для MSCATALOG. Это руководство всегда ссылается на таблицу как На MSCATALOG.

### Чтобы подключить базу данных INFORMIX из MicroStation:

1. Ввести с клавиатуры:

ACTIVE DATABASE <имя\_базы\_данных>

или

ОВ=<имя\_базы\_данных>

Или использовать диалоговое окно Соединение с базой данных

## Xbase

Следующие переменные конфигурации должны быть определены до соединения с базой данных Xbase из MicroStation.

Большинство этих переменных конфигурации находятся в категории База данных в диалоговом окне Конфигурация.

\$ (MSDIR) - сокращение для корневого каталога MicroStation.

Дополнительную информацию относительно определения переменных конфигурации смотри в разделе "Работа с переменными конфигурации" в главе 4 *Руководства пользователя*.

## dBASE IV (.mdx)

Назначение	Имя переменной	Определение
Файлы базы данных	MS_DBASE	\$ (MSDIR) database/xbase/
Загрузчик сервера	MS_SERVER	dbload
Связи базы данных	MS_LINKTYPE	XBASE (смотри "MS_LINKTYPE" на странице 12-25)
Загружаемые приложения проекта (категория Приложения проекта в диалоговом окне Конфигурация)	MS_DGNAPPS	Сервер
N/A (необязательно)	MS_USERMSC	Имя таблицы замещения для MSCATALOG. Это руководство всегда ссылается на таблицу как На MSCATALOG.

## dBASE III Plus (.ndx)

Назначение	Имя	Определение
------------	-----	-------------

переменной		
Файлы базы данных	MS_DBASE	\$ (MSDIR) database/ xbasendx/
Загрузчик сервера	MS_SERVER	db3ldlm.ma
Связи базы данных	MS_LINKTYPE	XBASE ("MS_LINKTYPE" на странице 12-25)

Назначение	Имя переменной	Определение
Загружаемые приложения проекта (категория Приложения проекта в диалоговом окне Конфигурация)	MS_DGNAPPS	Сервер
N/A (необязательно)	MS_USERMSC	Имя таблицы замещения для MSCATALOG. Это руководство всегда ссылается на таблицу как HaMSCATALOG.

## FoxBase (.cdx)

Назначение	Имя переменной	Определение
Файлы базы данных	MS_DBASE	\$ (MSDIR) database/xbasefox/
Загрузчик сервера	MS_SERVER	foxiddlm.ma
Связи базы данных	MS_LINKTYPE	XBASE ("MS_LINKTYPE" на странице 12-25)
Загружаемые приложения проекта (категория Приложения проекта в диалоговом окне Конфигурация)	MS_DGNAPPS	Сервер
N/A (необязательно)	MS_USERMSC	Имя таблицы замещения для MSCATALOG. Это руководство всегда ссылается на таблицу как На MSCATALOG.

Чтобы подключить базу данных Xbase из MicroStation:

1. Ввести с клавиатуры:

ACTIVE DATABASE <имя\_базы\_данных>

или

ОВ=<имя\_базы\_данных>

Или использовать диалоговое окно Соединение с базой данных.

### Примечания к DOS

Программа конфигурации MicroStation PC (для DOS) автоматически устанавливает требуемые определения переменных конфигурации, если выбрать пункт Xbase из меню опций Сервер БД. MS\_LINKTYPE (см. страницу 12-25) устанавливается в состояние записи связи с XBASE и чтения связей всех других типов. Дополнительную информацию о программе конфигурации смотри в разделе "Конфигурирование MicroStation" в главе 2 *Введения в MicroStation*.

Следующие переменные конфигурации должны быть определены до соединения со схемой RIS из MicroStation. Большинство этих



переменных конфигурации находятся в категории База данных в диалоговом окне Конфигурация. \$ (MSDIR) - сокращение для корневого каталога MicroStation. Дополнительную информацию относительно определения переменных конфигурации смотри в разделе "Работа с переменными конфигурации" в главе 4 *Руководства пользователя*.

Назначение	Имя переменной	Определение
Файлы базы данных	MS_DBASE	\$ (MSDIR) database/ris/
Загрузчик сервера	MS_SERVER	RISLDDLM.MA
Связи базы данных	MS_LINKTYPE	RIS
Загружаемые приложения проекта (категория Приложения проекта в диалоговом окне Конфигурация)	MS.DGNAPPS	Сервер
N/A (необязательно)	MS_USERMSC	Имя таблицы замещения для MSCATALOG. Это руководство всегда ссылается на таблицу как На MSCATALOG.
N/A (необязательно)	MS_DBFLOAT	Формат для чисел с плавающей запятой, возвращаемые как символьные строки символа. Значение по умолчанию - %. 7 д.
N/A (необязательно)	MS_DBDOUBLE	Формат для значений ДВОЙНОЙ ТОЧНОСТИ, возвращаемых как символьные строки. Значение по умолчанию - %.161д.

Чтобы подключить схему RIS из MicroStation:

1. Ввести с клавиатуры:

ACTIVE DATABASE <имя\_схемы> [.<паспорт>]

или

OB=<имя\_схемы> [.<паспорт>]

## MS LINKTYPE

Связи между элементами проекта и базой данных могут быть любыми из следующих допустимых типов:

- DMRS
- INFORMIX
- ORACLE
- ODBC
- RIS
- XBASE

Определение переменной конфигурации MS\_LINKTYPE управляет распознаванием типов связей программного обеспечения интерфейса баз данных с MicroStation.

DMRS - Data Management Retrieval System фирмы Intergraph. Интерфейс базы данных к DMRS обеспечивался в системе IGDS - предшественнике MicroStation. Можно установить MS\_LINKTYPE так, чтобы программное обеспечение интерфейса баз данных MicroStation распознавало связи DMRS.

### **Синтаксис**

Синтаксис для определения MS\_LINKTYPE имеет следующий вид:

MS\_LINKTYPE=<T<sup>TM</sup> \_связи> [; <тип\_связи>;...]

Например:

MS\_LINKTYPE=ORACLE; XBASE; INFORMIX

Первая спецификация типа - это тип связи, записанной, когда связь подключается к какому-либо элементу. Любые другие спецификации типа, следующие за первой, - это типы связей, которые MicroStation может "читать".

> • Чтобы изменить MS\_LINKTYPE без выхода из MicroStation:

1. В категории База данных диалогового окна Конфигурация отредактировать установку Связи с базой данных (MS\_LINKTYPE). Если необходимо, чтобы это изменение действовало после конца сеанса работы, то надо сохранить файл переменных конфигурации (смотри "Работа с переменными конфигурации" в главе 4 *Руководства пользователя*).

2. В окне ввода команд ввести с клавиатуры:

[reload ("["является вертикальной чертой, иногда называемой символом трубы.)

или

В диалоговом окне Окно SQL (открываемом при вводе с клавиатуры команды SQL) ввести следующую команду и затем щелкнуть кнопку Выполнить:

reload

Это повторно инициализирует типы связей.

Следующие разделы описывают дополнительные значения определения MS\_LINKTYPE. Дополнительную информацию о связях базы данных смотри в разделе "Создание связей" на странице 13-14.

Подключение связей к элементам

ORACLE - это тип связи по умолчанию. Если MS\_LINKTYPE определяет недопустимый тип связи и необходимо создать новую связь, то тип новой связи также будет ORACLE.

### **Просмотр атрибутов элементов в базе данных**

Нельзя просматривать связи, подключенные к какому-либо элементу, используя средство *Просмотр атрибутов элемента в базе данных*, если тип связи не определен в MS\_LINKTYPE (для чтения или записи).

<& Дополнительную информацию о просмотре связей смотри в разделе "Просмотр атрибутов" на странице 14-1.

### **Копирование элементов с подключенными связями**

При копировании элемента, имеющего связи с базой данных, связи базы данных копируются вместе с ним. Если копируемый элемент имеет тип связей, не определенный в MS\_LINKTYPE, то эта связь копируется независимо от того, какая связь должна

быть в текущий момент записана (первая связь в MS\_LINKTYPE), и независимо от текущей установки опции Режим связей.

Например, предположим, что скопирован элемент с подключенной к нему связью "ORACLE". Если MS\_LINKTYPE=DMRS, то связь, подключаемая к новому элементу (копия исходного элемента) будет той же самой связью с "ORACLE", которая была у исходного элемента, хотя тип связи Oracle не был определен в MS\_LINKTYPE. С другой стороны, если бы MS\_LINKTYPE=DMRS; ORACLE, то новый элемент имел бы подключенную связь - DMRS. При этом также учитывалась бы опция Режим связей, например, если меню опций Режим связей установлено в Новые, то связь типа DMRS будет установлена с новой строкой в базе данных.

Вышеупомянутый пример предполагает, что MicroStation соединена с базой данных. Если соединение отсутствует, то результат копирования элемента аналогичен ситуации, когда тип связи копируемого элемента не определен в MS\_LINKTYPE — то есть связь копируется независимо от MS\_LINKTYPE или меню опций Режим связей.

Дополнительную информацию о меню опций Режим связей смотри в разделе "Установка режима связей" на странице 13-17.

## Установка процедур загрузки экранных форм

Можно использовать **экранные формы** с интерфейсом базы данных MicroStation, чтобы форматировать атрибуты базы данных для просмотра и редактирования и для отображения атрибутов базы данных в проекте. Дополнительную информацию относительно этих операций смотри в разделах "Просмотр атрибутов" на странице 14 1 и "Отображение атрибутов базы данных в проекте" на странице 14-3.

Перед использованием программы работы с экранными формами, типа Oracle SQL\* Forms или INFORMIX PERFORM в MicroStation, необходимо создать процедуры загрузки экранных форм для "загрузки" значений в экранные формы.

**Триггер** - это SQL процедура, загружающая экранную форму с корректной информацией из базы данных при просмотре атрибутов или загрузке отображаемых атрибутов. В каждой создаваемой экранной форме необходимо создать соответствующий триггер.

Экранные формы PERFORM, используемые с MicroStation, состоят из двух частей. Первая часть - это описание экранной формы в файле "per" Вторая часть это программа INFORMIX 4GL, загружающая экранную форму, программа 4GL запускается из MicroStation

## Таблица MSFORMS

MicroStation создает и сопровождает таблицу, называемую MSFORMS, для передачи информации о связях базы данных с триггерами в экранных формах MSFORMS содержит одну строку с двумя полями, одно номер объекта и второе ключ mslink

Название поля	Тип
entity	integer
mslink	integer

Для каждой операции с формой имеется "целевая" строка в базе данных. Для операции просмотра атрибутов это строка таблицы, связанная с идентифицированным элементом. Для операции с отображаемыми атрибутами это строка таблицы, связанная с идентифицированным текстовым узлом. Перед каждой операцией с формами *MicroStation* просто обновляет номер объекта *entity* и ключ *mslink* в таблице *MSFORMS* для текущей целевой строки.

Когда какая-либо экранная форма была активизирована из *MicroStation*, то таблица *MSFORMS* была обновлена с помощью соответствующей строки. Следовательно, самая простая форма загрузки процедуры - это просто запрос к базе данных, основанный на ключе *mslink* из *MSFORMS*. Номер объекта *entity* в *MSFORMS* не используется в этой форме загрузки процедуры, поскольку экранные формы уже связаны с уникальными номерами объекта через *MSCATALOG*. Следовательно, экранная форма уже "знает" тип объекта и процедура загрузки может игнорировать номер объекта *entity*, найденный в *MSFORMS*.

В следующих разделах рассматриваются триггеры в примерах экранных форм Oracle SQL\* Forms, *parcel* (земельный участок) и *owner* (владелец), находящихся в каталоге "database/oracle/examples/forms". Аналогичные концепции применимы к примерам экранных форм INFORMIX, находящихся в каталоге *MicroStation's* "database/Informix/examples/forms".

Символ "/" используется для разделения каталогов в спецификациях пути. Фактический символ разделителя может быть другим для конкретной операционной системы.

#### **Чтобы создать таблицы и индексные файлы для упражнений:**

1. Используя SQLPlus, SQL\* DBA или другую графическую утилиту SQL, в системной командной строке ввести:

```
SQL pl us
```

2. Выполнить командные файлы "mscat.sql" и "gis.sql". В приглашении SQL> ввести:

```
@database/oracle/mscat.sql
```

```
@database/oracle/examples/tables/gis.sql
```

3. Выйти из SQLPlus.

#### **Чтобы загрузить данные для примеров в таблицы:**

1. Установив каталог *MicroStation* "database/oracle/examples/tables" как текущий каталог, ввести в системной командной строке:

```
loadgis <имя_пользователя> <паспорт>
```

#### **Чтобы создать файлы ".frm" используемые для отображения экранных форм в Oracle:**

1. Установив каталог *MicroStation* "database/oracle/examples/forms" как текущий каталог, ввести в системной командной строке:

```
makeform <имя_пользователя> / <паспорт>
```

#### **Чтобы установить переменную конфигурации MS\_DBASE для путей к файлам упражнений Oracle:**

1. В категории База данных диалогового окна Конфигурация выбрать Файлы базы данных (MS\_DBASE).

2. Щелкнуть кнопку Выбрать.

- Откроется диалоговое окно Выбор пути для файлов базы данных.
3. Щелкнуть кнопку Добавить.
  4. В поле списка каталогов выбрать путь, который соответствует следующей спецификации:  
\$ (MSDIR) database/oracle/ examples/forms/
  5. Щелкнуть кнопку Добавить.
  6. Щелкнуть кнопку Выполнено.
  7. (Необязательно) Если необходимо, чтобы это изменение осталось действующим после конца сеанса работы, то выбрать пункт Сохранить из меню Файл в диалоговом окне Конфигурация.
  8. Щелкнуть кнопку ОК в диалоговом окне Конфигурация или ввести -wdoracle в командной строке.

## INFORMIX

Примеры таблиц базы данных для INFORMIX находятся в каталоге MicroStation "database/informix/ examples/tables".

### Общая процедура настройки для упражнений INFORMIX

1. Отредактировать первую строку командного файла SQL "gis.sql". Заменить knox. 1 од на полный путь к этому файлу — например, /usr/tom/i'nformix/knox.log.
2. Выполнить "gis.sql", чтобы создать необходимые таблицы и индексные файлы, включая управляющую таблицу MS CATALOG и загрузить данные для упражнений. Установив каталог MicroStation "database/informix/examples/gis" как текущий каталог, ввести в системной командной строке:  
  
isql - gis.sql
3. Включить каталог MicroStation "informix/examples/forms" в определение системной переменной INFORMIX DBPATH. Этот каталог содержит примеры экранных форм PERFORM.
4. Запустить MicroStation и проверить, что переменные конфигурации установлены правильно для использования интерфейса базы данных. Смотри таблицу "INFORMIX" на странице 12-19. Необходимо также проверить, что тип связи DMRS включен для чтения в MS\_LINKTYPE (см. страницу 12-25). Файлы проекта для упражнений имеют элементы со связями типа DMRS.
5. Добавить спецификации пути для каталогов, которые содержат примеры файлов базы данных в определение переменной конфигурации (MS\_DBASE).
6. Подключиться к базе данных. Смотри "Чтобы подключить базу данных INFORMIX из MicroStation:" на странице 12-20.  
  
Символ "/" используется для разделения каталогов в спецификациях пути. Фактический символ разделителя может быть другим для конкретной операционной системы.  
  
Чтобы установить переменную конфигурации MS\_DBASE для пути к файлам упражнений Informix:

1. В категории База данных диалогового окна Конфигурация выбрать Файлы базы данных (MS\_DATABASE).
  2. Щелкнуть кнопку Выбрать.
- Откроется диалоговое окно Выбор пути для файлов базы данных.
3. Щелкнуть кнопку Добавить.
  4. В поле списка каталогов выбрать путь, который соответствует следующей спецификации:  
\$ (MSDIR) database/mformix/examples/fbrms/
  5. Щелкнуть кнопку Добавить.
  6. Щелкнуть кнопку Выполнено.
  7. (Необязательно) Если необходимо, чтобы изменение осталось действующим после конца сеанса, то выбрать пункт Сохранить из меню Файл в диалоговом окне Конфигурация.
  8. Щелкнуть кнопку ОК в диалоговом окне Конфигурация.
- Примеры файлов базы данных для Xbase устанавливаются следующим образом:

Формат индексного файла	Каталог файлов примеров
dBASE IV (.mdx)	\$ (MSDIR) database/xbase/examples/gis
dBASE III Plus (.ndx)	\$ (MSDIR) database/xbasendx/examples/gis
FoxBase (.cdx)	\$ (MSDIR) database/xbasefox/examples/gis

## Общая процедура настройки для упражнений с Xbase

1. Запустить MicroStation и проверить, что переменные конфигурации установлены правильно для использования интерфейса базы данных. Смотри таблицу "Xbase" на странице 12-20. Необходимо также проверить, что тип связи DMRS включен для чтения в MS\_LINKTYPE (см. страницу 12-25). Файлы проекта для упражнений имеют элементы со связями типа DMRS.
  2. Добавить спецификации пути для каталогов, которые содержат примеры файлов базы данных в определение переменной конфигурации (MS\_DATABASE).
- Словарь данных уже был инициализирован для примеров XBase.
3. Выполнить командный файл "build.bat", чтобы сформировать словарь данных для упражнения.
  4. Подключиться к базе данных gis. В окне команд ввести с клавиатуры:  
DB=gis  
ИЛИ  
Использовать меню Установки>База данных>Соединить.
- Программа конфигурации MicroStation PC (для DOS) автоматически устанавливает требуемые определения переменных конфигурации (шаги 1-2), если выбрать Xbase из меню опций Сервер БД.

## Триггер KEY-STARTUP Oracle

Имеются фактически два триггера в экранной форме *parcel* (земельный участок), которые могут использоваться для просмотра атрибутов и операций редактирования. Эти триггера являются универсальными (а не специальными для таблиц земельного участка или конкретного упражнения). Поведение экранной формы зависит от того, активизировалась ли она из *MicroStation*.

Первый шаг включает триггер KEY-STARTUP. Триггера KEY-STARTUP активизируются при запуске экранной формы. Триггер KEY-STARTUP имеет следующий вид:

```
DECLARE
localMslink NUMBER;

BEGIN
: GLOBAL.isMsformsValue = 'NO';

select mslink into localMslink from msforms;

IF localMslink <> 0 THEN
: GLOBAL.isMsformsValue = 'YES';

EXECUTE_QUERY;

END IF;

END;
```

Этот триггер определяет, была ли экранная форма активизирована из *MicroStation*, и действует в соответствии с этим. Если эта форма была активизирована из *MicroStation*, то *local Mslink* имеет ненулевое значение и запрос выполняется точно так же, как если бы пользователь нажал клавишу <Enter Query> в интерактивном сеансе. Если форма не была активизирована из *MicroStation*, то *localMslink* - ноль, а форма запускается без ввода режима запроса.

## Oracle PRE-QUERY триггер

В случае, если форма была активизирована из *MicroStation*, то необходим триггер PRE-QUERY, который выполняется перед исполнением любого запроса. Таким образом, если значение *mslink* в *MSFORMS* отлично от нуля, то выполняется запрос и активизируется этот триггер. Триггер PRE QUERY имеет следующий вид:

```
IF: GLOBAL.isMsformsValue = 'YES' THEN
select mslink into: mslink from msforms;

END IF;
```

Если форма была активизирована из *MicroStation*, то значение *mslink* из *MSFORMS* вставляется в поле *mslink* в форме до запроса. В противном случае никаких действий не выполняется. Следует обратить внимание, что экранная форма должна содержать столбец *mslink* для этого метода. Необходимо также отметить, что любые обновления информации в форме, обновляют основную строку в базе данных.

## Oracle PRE-FORM триггер

Другой метод использования триггера полезен для форм, не содержащих столбец *mslink*. В экранной форме *owner* (владелец), используемой для отображения атрибутов базы данных в проекте, триггер PRE-FORM выбирает несколько столбцов из таблицы *parcel* (земельный участок) и помещает их в поля этой формы. Триггер PRE-FORM имеет следующий вид:

```
select owner, value, house_num, street_name,
city, state, zip_code into: owner.: value,
.hoLise_num,: street_name, -city.: state,
: zip_code from parcel where mslink
(select mslink from MSFORMS);
```

В отличие от триггеров в ра reel (земельный участок), триггер PRE-FORM в owner (владелец) должен быть изменен при использовании с другой базой данных.

## Подключение строк базы данных к элементам

В этой главе описаны процедуры связывания информации в базе данных к элементам в MicroStation.

### Использование SQL

SQL (произносится эс-ку-эль) означает Structured Query Language (структурированный язык запросов). SQL простой и мощный язык, который является промышленным стандартом для манипулирования данными и доступа к базе данных. Предложения SQL могут использоваться в интерактивном режиме для запросов к базе данных непосредственно из MicroStation.

Поддерживаются все предложения SQL, распознаваемые программным обеспечением базы данных, включая:

SELECT	Запрос к таблицам базы данных.
UPDATE	Редактирование строки в таблице.
DELETE	Удаление строк.
INSERT	Добавление новых строк.
CREATE	Определение новой таблицы и добавление ее к базе данных.
DROP	Удаление существующей таблицы.

Синтаксис предложений SQL описывается в разделе "Предложения SQL" в главе 17, "Описание интерфейса базы данных". *Справочного руководства.*

### Использование окна Окно SQL

Диалоговое окно Окно SQL используется для выполнения предложений SQL и просмотра результатов.

> • Чтобы открыть окно Окно SQL:

• Ввести с клавиатуры SQL.

Окно SQL имеет большую секцию сверху для просмотра запросов SQL и небольшую секцию внизу для передачи на выполнение предложений SQL.





Чтобы выполнить SQL предложение:

1. В окне Окно SQL набрать SQL предложение. Предложение можно вводить непрерывной строкой с автоматическим переносом слов или можно нажимать <Return> в конце каждой строки.

Литеральные выражения, используемые в назначениях и сравнениях могут иметь символьный (char) или числовой (number) формат или формат даты (последний, только в Xbase). Символьные строки разделяются одиночной кавычкой. Выражения даты для Xbase имеют следующий вид:

DATE ('07/04/76'). Следует отметить необходимость записи в формате даты предшествующих нулей.

2. Щелкнуть кнопку Выполнить.

Запрошенная информация базы данных выводится построчно. Можно использовать полосу прокрутки, если длина строки превышает размер окна.

3. (Необязательно) Чтобы перейти к следующей строке, щелкнуть кнопку Следующая.

Допустимо следующее предложение SELECT:

```
SELECT* FROM mytable WHERE mychar = 'Character' AND
myfloat = 3.5 AND myint = 4 AND mydate = DATE (07/04/
76)
```

## Командные файлы

Для удобства часто используемое предложение SQL можно сохранять в **командном файле**. Командный файл - это текстовый файл, в котором записано предложение SQL. Хотя файл может содержать только одно предложение, это предложение может продолжаться на многих строках. Расширение имени файла по умолчанию - ".sql."

Некоторые команды интерфейса базы данных определяют предложения SQL, как например, ACTIVE ENTITY (AE=). В отличие от фактических предложений SQL эти команды не могут быть выполнены в окне Окно SQL; они должны вводиться с клавиатуры в окне ввода команд. Подобно всем командам MicroStation, вводимым с клавиатуры, эти команды могут также быть активизированы из командного файла командой @<командный\_файл>. (Расширение имени файла нельзя опускать.) Смотри "Сценарии команд" на странице 8-1 *Руководства пользователя*.

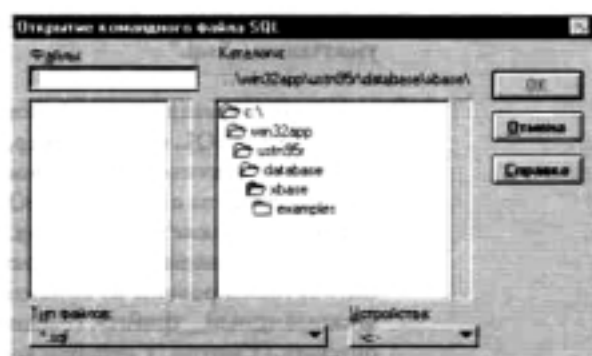
### > • Чтобы создать командный файл:

1. В окне Окно SQL ввести предложение SQL, которое необходимо сохранить.

2. Из меню Файл окна Окно SQL выбрать пункт Сохранить как. Откроется диалоговое окно Сохранить командный файл SQL.
3. В поле имени ввести с клавиатуры желаемое имя файла.
4. Установить желаемый каталог.
5. Щелкнуть кнопку ОК.

**> • Чтобы выполнить предложение SQL, сохраненное в командном файле:**

1. Из меню Файл окна Окно SQL выбрать пункт Открыть. Откроется диалоговое окно Открыть командный файл SQL.



2. Выбрать файл, содержащий желаемое предложение.
  3. Щелкнуть кнопку ОК.
- ИЛИ

• В окне Окно SQL выполнить @<имя\_файла>. Расширение файла по умолчанию - \*.sql".

## Передача на выполнение предложений SQL вручную

Предложения SQL генерируются MicroStation каждый раз при использовании некоторых инструментальных средств базы данных и команд, вводимых с клавиатуры. Например, MicroStation генерирует предложение SQL - INSERT при использовании средства *Подключение активного объекта* и меню опций Режим связей, установленном в Новые (смотри "Установка режима связей" на странице 13-17).

По умолчанию, MicroStation немедленно передает на выполнение предложения SQL программному обеспечению базы данных. В альтернативном режиме MicroStation передает на выполнение предложения только при выполнении действий, требующих добавления строки к таблице базы данных. Процесс добавления строк требует, чтобы программное обеспечение базы данных заблокировало таблицы, и для снятия блокировки нужно осуществить передачу на выполнение предложения SQL. Альтернативный режим может быть полезен, когда необходимо иметь возможность отмены невыполненных предложений SQL.

Интерфейс базы данных Xbase не поддерживает альтернативный режим, описанный выше. Смотри "Обработка SQL предложения" на странице 13-21.

**Чтобы активизировать режим, в котором MicroStation не передает автоматически на выполнение предложения SQL:**

1. В окне ввода команд ввести с клавиатуры:

```
SESSION AUTOCOMMIT OFF
```

**Чтобы явно потребовать от MicroStation передачи на выполнение предложений:**

1. В окне Окно SQL выполнить предложение:

COMMIT

Чтобы отменить действие всех последних предложений:

1. В окне Окно SQL выполнить предложение:

ROLLBACK

Чтобы вновь активизировать режим по умолчанию:

1. В окне ввода команд ввести с клавиатуры:

SESSION AUTOCOMMIT ON

## Создание таблиц

Создание таблицы - это просто выполнение стандартной команды SQL - CREATE TABLE.

**Чтобы создать таблицу:**

1. В окне Окно SQL выполнить следующее предложение:

```
CREATE TABLE имя_таблицы  
(Column_name_1 Column_spec_1,  
Column_name_2 Column_spec_2,
```

Спецификация Column (Столбец) может быть любого типа, поддерживаемого SQL, включая char, numeric и date.

Единственным требованием для активизации строки, связанной с элементом, является обязательное наличие в таблице столбца с именем mslink.

Например, предположим, что в примере карты, на которой основано упражнение в начале этой главы, необходимо создать новую таблицу с целью оценки налогов на недвижимость. Для этого можно было бы выполнить следующее предложение CREATE TABLE:

```
CREATE TABLE assessor  
(clt_no char (30),  
as_dt date,  
as_val number (10),  
tax_rate number (10, 4),  
mslink number (10))
```

В результате была бы создана следующая таблица (данные приведены для примера): Следует обратить внимание на общий столбец - clt\_no для новой таблицы и уже существующей таблицы parcel (земельный участок).

assessor (налоговый инспектор) clt_no	as_dt	as_val	tax_rate	mslink
119-LA9	05/17/91	180,000	0.0500	37
119-LB36	03/17/92	144,000	0.0500	38
119-LB35	08/23/92	180,000	0.0500	39

parcel (участок) mslink Clt_no	house_number	name	val ne	area
1 119 LA9	220	BRIDGEWATER RD	9000	16,036
14 119LB36	8717	KILDARE DR	7200	16,668
15 119LB35	8713	KILDARE DR	9000	16,306

## Столбец Mslink

Столбец mslink служит первичным ключом таблицы. Это означает, что любая строка может быть однозначно определена исключительно значением столбца mslink.

## Индекс ключа

При выполнении предложений SELECT, UPDATE или DELETE осуществляется поиск заданной таблицы. Индексы используются всякий раз, когда это возможно, чтобы ускорить поиск. Индекс - это связанный список, который содержит только интересующий столбец и номер его строки в таблице. Использование индексов необязательно, но они значительно повышают эффективность работы с базой данных. Любой столбец в таблице может иметь собственный индекс. Конечно, каждый индекс требует дополнительной дисковой памяти, так что необходимо найти компромисс между размером файлов и эффективностью

Так как столбец msl i n k это первичный ключ таблицы, имеет смысл создание индекса mslink. Следует создавать этот индекс, используя Вашу программу базы данных.

## Таблица MSCATALOG

Прежде чем затратить много времени на создание таблиц, следует хорошо разобраться с назначением таблицы MSCATALOG (иногда называемой управляющей таблицей). Это специальная таблица, сообщающая MicroStation информацию о базе данных. Она открывается при каждом соединении базы данных и содержит строку для каждой таблицы, связанной с файлом проекта. Имеется только одна таблица MSCATALOG для каждой базы данных.

Любая таблица в какой-либо базе данных, связанная с элементами, должна иметь соответствующую строку в MSCATALOG. Однако при использовании предложений SQL возможен доступ из MicroStation к любой таблице в базе данных, даже если не имеется соответствующей строки в MSCATALOG. Поскольку невозможно подключиться к базе данных, которая не имеет таблицы с именем MSCATALOG, то следует создавать ее даже, если не планируется связь таблицы с элементами.

## Структура таблицы Mscatalog

Таблица MSCATALOG имеет следующую структуру:

Столбец	Тип (длина)	Описание
tablename	char (32)	имя таблицы
entitynum	number (10)	уникальный номер объекта или таблицы
screen-form	char (64)	имя экранной формы
reporttable	char (64)	имя таблицы отчетов
sql review	char (240)	запрос просмотра SQL по умолчанию
fencefilter	char (240)	фильтр выделенной области SQL по умолчанию
dastable	char (32)	имя таблицы отображаемых атрибутов

formtable	char (64)	таблица, используемая для локализации экранной формы (необязательно)
nextocc	(только интерфейс базы данных RIS)	(Смотри "Столбец nextocc MSCATALOG" на странице 13-21)

#### **столбец tablename**

Столбец tablename в MSCATALOG содержит имена таблиц в базе данных, которые должны быть связаны с элементами. Имена таблиц должны быть добавлены в tablename до установления связи с элементами.

#### **столбец entitynum**

Столбец entitynum в строке MSCATALOG для определенной таблицы содержит номер объекта в таблице. Номер объекта - это положительное целое число от 1 до 65535, которое однозначно идентифицирует таблицу в базе данных; каждая таблица, используемая с каким-либо файлом проекта, должна иметь уникальный номер объекта.

#### **столбец screenform**

Столбец screenform в строке MSCATALOG для заданной таблицы определяет экранную форму по умолчанию, используемую вместе со средством *Просмотр атрибутов элемента в базе данных* для просмотра атрибутов элементов, связанных со строками в таблице базы данных. Если в MSCATALOG существует необязательный столбец formtable (см. страницу 13-10), то просматривается этот столбец для локализации экранной формы, которую нужно использовать вместо экранной формы по умолчанию. Если столбец formtable пуст, то будет использоваться столбец screenform для просмотра и редактирования.

#### **столбец reporttable**

Столбец reporttable в строке MSCATALOG для заданной таблицы содержит имя таблицы отчетов, которая может быть создана средством *Генерация таблицы отчетов* для подготовки отчета по атрибутам элементов, связанных со строками в таблице базы данных. Этот столбец обычно загружается командой ACTIVE REPORT (RS=) (см. главу 17, "Описание интерфейса базы данных". *Справочного руководства*).

#### **столбец sqlreview**

Столбец sqlreview строке MSCATALOG для заданной таблицы содержит предложение SQL - SELECT, которое выполняется со средством *Просмотр атрибутов элемента в базе данных* для просмотра атрибутов элементов, связанных со строками в таблице базы данных. Этот столбец обычно загружается командой ACTIVE REVIEW (RA=) (см. главу 17, "Описание интерфейса базы данных", *Справочного руководства*).

#### **столбец fencefilter**

Столбец fencefilter в строке MSCATALOG для заданной таблицы содержит предложение SQL - SELECT которое определяет какие элементы, связанные со строками в таблице, выбраны для операций с выделенной областью. Этот столбец обычно загружается командой DEFINE SEARCH (DS=) (см. главу 17, "Описание интерфейса базы данных", *Справочного руководства*).

#### **столбец dastable**

Столбец dastable в строке MSCATALOG для заданной таблицы определяет таблицу отображаемых атрибутов, используемую для

определения, какие атрибуты элементов, связанных со строками таблицы, будут отображаться в текстовых узлах отображаемых атрибутов, и определяет используются ли экранные формы для отображения атрибутов.

Каждая строка в таблице отображаемых атрибутов (смотри следующую таблицу) определяет номер типа отображаемого атрибута и его определение.

Столбец	Тип (длина)	NULL
dastype	number (3)	нет
formname	char (240)	да
sqldas	char (240)	да

Столбец formname для специфического типа отображаемого атрибута определяет экранную форму, используемую для отображения атрибутов. Столбец sqldas определяет предложение SQL - SELECT, выбирающее атрибуты для отображения. Для получения дополнительной информации о текстовых узлах отображаемых атрибутов смотри "Отображение атрибутов базы данных в проекте" на странице 14-3.

#### **столбец formtable**

Необязательный столбец formtable в строке MSCATALOG для заданной таблицы определяет таблицу форм, используемую для локализации экранных форм для просмотра и редактирования атрибутов элементов, связанных со строками в таблице базы данных.

Таблица форм имеет следующую структуру:

Столбец	Тип (длина)	NULL
tablename	char (32)	НЕТ
type	char (64)	НЕТ
formname	char (64)	НЕТ

Столбец type (тип) может иметь значение review или edit, но они должны быть записаны строчными буквами.

При использовании средства *Просмотр атрибутов элемента в базе данных*, если меню опций Формы установлено в Текстовые формы в установочном окне База данных, то экранная форма локализуется при первом просмотре столбца formtable в MSCATALOG для соответствующей таблицы. Если таблица формы

определена и существует, то в ней осуществляется поиск имени используемой формы. Если найдено имя формы для таблицы текущей связи и ее тип - review, то используется именно эта форма. Однако, если редактируется строка (или редактируется активный объект командой EDIT AE), то используется форма типа edit.

Если таблица формы не найдена или не существует, то для операций просмотра и редактирования используется форма, определенная в столбце screen-form MSCATALOG.

## **Создание и сопровождение MSCATALOG**

Предложение SQL для создания MSCATALOG находится в командном файле SQL - "mscat.sql."

> • **Чтобы создать MSCATALOG:**

1. Из подменю База данных меню Установки выбрать пункт Настройка.

Откроется окно предупреждения с вопросом о необходимости создания MSCATALOG.

2. Щелкнуть кнопку Да.

Создается MSCATALOG и открывается установочное окно MSCATALOG.

ИЛИ

1. В Окне SQL выполнить:

@>mecat

Вышеупомянутая процедура не работает с интерфейсом базы данных Xbase. При использовании Xbase можно выполнить соответствующее предложение CREATE TABLE или непосредственно создать MSCATALOG, используя программное обеспечение Вашей базы данных.

**> • Чтобы создать таблицу в базе данных:**

1. Из подменю База данных меню Установки выбрать пункт Настройка.

Откроется установочное окно MSCATALOG.

2. Из меню Таблица этого диалогового окна выбрать пункт Создать.

Откроется диалоговое окно Создание таблицы.

3. В поле Имя таблицы ввести имя таблицы.

4. Щелкнуть кнопку Вставить.

Откроется диалоговое окно Вставка поля.

5. В поле Имя поля ввести имя первого столбца.

6. Из меню опций Тип выбрать тип данных этого столбца — Символьн., Целое, Действит. или Дата.

7. Если опция Тип - Символьн.(ый), то ввести длину поля (число символов) в поле Ширина.

или

Если опция Тип - Действит.(ельное), то ввести общее количество цифр в поле Ширина и число цифр после десятичной точки в поле Десятичн.(ый).

или

Иначе, перейти к шагу 8.

8. Щелкнуть кнопку ОК, чтобы вернуться к диалоговому окну Создание таблицы.

9. Для каждого дополнительного столбца повторить шаги 4-8. (Чтобы отредактировать или удалить предварительно определенный столбец, следует использовать кнопки Изменить и Удалить.)

10. Щелкнуть кнопку ОК.

**• Чтобы добавить таблицу к MSCATALOG:**

1. Из подменю База данных меню Установки выбрать пункт Настройка.

Откроется установочное окно MSCATALOG.

2. Из меню Таблица диалогового окна выбрать Добавить в MSCATALOG.

Откроется диалоговое окно Добавить в MSCATALOG.

3. Из поля списка Таблицы выбрать добавляемую таблицу. Это станет значением поля `tablename` в строке `MSCATALOG` для соответствующей таблицы.

4. В поле Номер таблицы ввести уникальный номер объекта для данной таблицы. Это станет значением поля `entitynum` в строке `MSCATALOG` для соответствующей таблицы.

5. Щелкнуть кнопку ОК.

#### **Чтобы удалить таблицу из MSCATALOG:**

1. Из подменю База данных меню Установки выбрать пункт Настройка.

Откроется установочное окно `MSCATALOG`.

2. В поле списка Таблицы выбрать удаляемую таблицу.

3. Из меню Таблица диалогового окна выбрать пункт Удалить из `MSCATALOG`.

#### **Чтобы удалить таблицу из базы данных:**

1. Из подменю База данных меню Установки выбрать пункт Настройка.

Откроется установочное окно `MSCATALOG`.

2. Из меню Таблица диалогового окна выбрать пункт Удалить. Откроется диалоговое окно Удалить таблицу.

3. Из поля списка Таблицы выбрать удаляемую таблицу.

4. Щелкнуть кнопку ОК.

#### **Чтобы отключить базу данных:**

- Из подменю База данных меню Установки выбрать пункт Отсоединить.

Установочное окно База данных открывается при выборе пункта Диалог из подменю База данных меню Установки.

#### **Чтобы добавить новую строку (для новой таблицы) в MSCATALOG:**

1. В окне Окно SQL передать на выполнение:

```
INSERT INTO MSCATALOG VALUES
('<имя_таблицы>', <номер_объекта>,
'<экранная_форма>', '<имя_таблицы1_отчетов>', "", "", "")
или
```

```
INSERT INTO MSCATALOG (tablename, entitynum, screenform,
reporttable) VALUES
('<имя_таблицы>', <номер_объекта>,
'<экранная_форма>', '<имя_таблицы_отчетов>')
```

При использовании первого варианта должны быть вставлены все столбцы; следует заметить, что последние четыре значения пусты.

2. В окне Окно SQL передать на выполнение:

```
RELOAD
```

Например, чтобы вставить строку в `MS CATALOG` для примера таблицы `assessor` (налоговый инспектор), можно было бы выполнить (для шага I):

```
INSERT INTO MSCATALOG VALUES
('assessor', 4, 'as_screen', 'as_report', "", "", "")
```

## **Инструментальные средства базы данных**



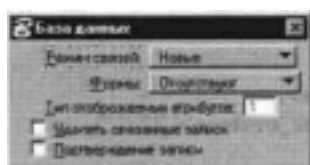
В разделе "Команды с клавиатуры интерфейса базы данных" в главе 17, "Описание интерфейса базы данных". *Справочного руководства* описываются инструментальные средства для использования интерфейса базы данных. Эта инструментальная панель используется так же как любая другая инструментальная панель MicroStation. Чтобы открыть инструментальную панель База данных:

- Из меню Инструменты выбрать пункт База данных.

## Создание связей

Для создания связей элементов в файле проекта к строкам в базе данных используется инструментальное средство *Подключение активного объекта* в инструментальной панели База данных.

## Общая процедура создания связи (ей)



1. Определить активный объект (смотри "Определение активного объекта" на странице 13-15.)
2. В установочном окне База данных установить меню опций Режим связей (смотри "Установка режима связей" на странице 13-17.)
3. Использовать средство *Подключение активного объекта* для подключения элемента (ов) к строке, обозначенной как активный объект

## Определение активного объекта



Строка, обозначенная как целевая при создании связи, называется **активный объект**. Она становится единственной строкой в специальной таблице с именем AE. Активный объект можно представить как буферную область для размножения данных. Он может быть создан несколькими различными путями:

- Он может быть определен графически с помощью средства *Графическое определение активного объекта*.
- Он может быть создан непосредственно командой ACTIVE ENTITY (AE=), вводимой с клавиатуры.
- Он может быть локализован и скопирован из строки прототипа в существующей таблице командой FIND (Fi=), вводимой с клавиатуры.

**Чтобы непосредственно определить активный объект командой, вводимой с клавиатуры •**

1. В окне ввода команд ввести:

AE=INSERT INTO <имя\_таблицы>

(<column\_1>, <column\_2>,...)

VALUES (<value\_1>, <value\_2>...)

или

ACTIVE ENTITY INSERT INTO <имя\_таблицы>

(<column\_1>, <column\_2>,...)

VALUES (<value\_1>, <value\_2>...)

Здесь создается активный объект в таблице AE с той же самой структурой данных как и в таблице с именем имя\_таблицы.

**Чтобы определить активный объект по строке прототипа:**

1. В окне ввода команд ввести:

FI=SELECT\* FROM <имя\_таблицы> WHERE



<column\_name>='<value>'

FIND SELECT\* FROM <имя\_таблицы> WHERE

<column\_name>='<value>'

Для проверки значений в активном объекте используется инструментальное средство *Отображение активного объекта*.

### **Редактирование активного объекта**

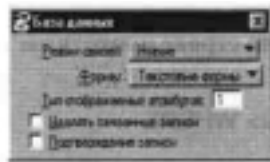
Можно отредактировать активный объект из MicroStation перед установлением связи его с каким-либо элементом. Для этого можно использовать экранную форму или предложение SQL UPDATE.

Команда EDIT AE используется с экранными формами. Если меню опций Формы установлено в Текстовые формы в установочном окне База данных, то имя экранной формы для редактирования выбирается из таблицы, определенной в необязательном столбце formtable MSCATALOG. Тип этой экранной формы может быть review или edit. Если formtable не имеет конкретного значения, то используется строка screenform.

Если экранная форма не определена или если меню опций Формы установлено в Нет, то активный объект надо редактировать, используя команду SQL UPDATE.

*Its* В зависимости от меню опций Режим связей, которое устанавливается в установочном окне База данных, редактирование активного объекта может изменять строку в базе данных. Если меню опций Режим связей установлено в Новые, то активный объект - это запись прототип; редактирование активного объекта не воздействует на какие-либо строки. Однако, если меню опций Режим связей не установлено в Новые, то редактирование активного объекта изменяет соответствующую строку. Для получения дополнительной информации, смотри "Установка режима связей" на странице 13-17.

- • **Чтобы отредактировать активный объект, используя экранную форму:**



1. Из подменю База данных меню Установки выбрать Диалог. Откроется установочное окно База данных.
2. Из меню опций Формы выбрать Текстовые формы.
3. В окне команд ввести с клавиатуры:

EDIT AE

Выводится экранная форма для редактирования.

*Экранная форма для редактирования, как она выглядит в базе данных Xbase. В базе данных Oracle форма изображается в текстовом окне.*

4. Отредактировать активный объект.

Если экранная форма имеет более одной страницы, то нажать <PgDn> или щелкнуть кнопку Page Down, чтобы перейти на другую страницу.

5. Щелкнуть кнопку OK.

**Чтобы отредактировать активный объект, используя команду SQL UPDATE:**

1. В окне Окно SQL выполнить:

UPDATE AE SET <столбец> = <значение>,  
<столбец> = <значение>,...

Например, для таблицы parcel (земельный участок) можно было бы выполнить:

1. UPDATE AE SET house\_num= '8716',  
str\_name = 'KILDARE DR', val ue = 8000

## Установка режима связей

**Режим связей**, который устанавливается в установочном окне База данных, определяет, как обрабатывается активный объект при установлении связи элемента с ним. Если меню опций Режим связей установлено в Отсутствуют, то никакие связи не могут быть созданы.

### Режим связи Новые

**Режим связи Новые** используется в том случае, если каждой связи должна соответствовать уникальная строка, как, например,

в случае участков земли на карте. В режиме связи Новые активный объект фактически является "строкой прототипа", которая не должна полностью соответствовать существующей строке в таблице.

При создании какой-либо связи в режиме Новые копия активного объекта добавляется к таблице как новая строка. Затем элемент связывается с этой новой строкой. Если связь элементов к базе данных осуществляется исключительно в режиме Новые, то можно гарантировать, что каждый связываемый элемент будет иметь соответствующую уникальную строку в базе данных.

#### **Режим связи Дублирующие**

**В режиме связи Дублирующие** каждый обозначенный элемент связывается с одной и той же строкой, пока эта строка остается активным объектом. Это полезно, когда необходимо связать элементы только с "общей родовой строкой" вместо соединения с отдельным экземпляром этой строки.

Например, при проектировании печатной платы неудобно связывать с отдельными строками большое число электрических компонентов таких, как резисторы и конденсаторы. Точный подсчет числа компонентов возможен, даже если резисторы одного типа будут "совместно использовать" строки базы данных.

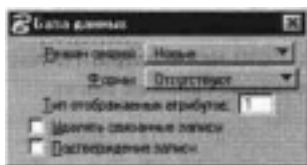
#### **Режим связи Информационные**

**Режим связи Информационные** идентичен режиму Дублирующие за исключением того, что специальный бит устанавливается для каждой связи. Это различие важно для приложений, проверяющих этот бит при обработке связей. Связи, созданные в режиме Информационные, не могут использоваться при подготовке отчетов и не добавляют (удаляют) строки таблицы при копировании связанного элемента.

##### **• Чтобы установить режим связей:**

1. Из подменю База данных меню Установки выбрать Диалог. Откроется установочное окно База данных.
2. Из меню опций Режим связей выбрать требуемый режим — Отсутствуют, Новые, Информационные или Дублирующие.

## **Подключение связей к библиотекам фрагментов**



Прикладные программы часто нуждаются в связывании информации базы данных с фрагментами. Для связывания библиотеки фрагментов с существующей таблицей базы данных используется установочное окно Связи фрагмента с БД, которое подключает связи базы данных к фрагментам библиотеки. Это делает возможным размещение фрагментов в проекте с уже установленной связью с базой данных. Для получения дополнительной информации об установочном окне Связи фрагмента с БД, смотри "Установочное окно Связи фрагментов с БД" в главе 17, "Описание интерфейса базы данных", *Справочного руководства*.

## Примечания к Oracle

Этот раздел относится к использованию интерфейса базы данных Oracle

### Редактирование активного объекта

Если меню опций Режим связей установлено в Новые, то команда EDIT AE может использоваться только для добавления данных к активному объекту, а не для редактирования существующих данных.

## Примечания к INFORMIX

Этот раздел относится к использованию интерфейса базы данных INFORMIX

### Тип ANSI

Любая база данных INFORMIX, используемая с MicroStation, должна быть типа ANSI.

### Тип данных mslink

Столбец mslink может иметь типы INTEGER или SERIAL. Если столбец mslink типа SERIAL, то INFORMIX вместо MicroStation назначит уникальный последовательный ключ каждой строке, добавленной к таблице.

### Редактирование активного объекта

Смотри "Редактирование активного объекта" на странице 13-19.

## Примечания к Xbase

Этот раздел относится к использованию интерфейса баз данных Xbase.

### Многие пользователи

Словарь данных содержит информацию о владельце объекта базы данных. Однако, Xbase не поддерживает понятие пространства пользователя. Чтобы избежать конфликтов между различными пользователями одной базы данных, следует использовать MicroStation с выключенным (по умолчанию) предпочтением пользователя. Использовать единые таблицы AE/MSFORMS (см. главу 17, "Описание интерфейса базы данных". *Справочного руководства*).

### Индексные файлы

При выполнении предложений SELECT, UPDATE или DELETE осуществляется поиск определенной таблицы. Интерфейс Xbase MicroStation может одновременно использовать только один индекс на таблицу. Кроме того индекс не может строиться на выражении, включающем операцию ИЛИ.

Не принимается никаких специальных мер, чтобы выбрать индекс, повышающий эффективность. Если для данной таблицы оценивается некоторое множество условий, то используется индекс первого встреченного индексированного выражения. Чтобы повысить эффективность поиска, следует внести первым в список условие с наиболее полезным индексом.

## Выражения с Date в предложениях SQL

Специальный формат Date разрешается в литеральных выражениях внутри предложений SQL, выполняемых из MicroStation. Пример этого формата: DATE ('07/04/76'). Предшествующие нули обязательны.

## Обработка SQL предложения

В случае SQL баз данных, типа Oracle, предложения SQL, выполняемые из MicroStation, передаются базе данных для обработки, а результаты возвращаются обратно. Xbase отличается тем, что обработка SQL выполняется внутри MicroStation. Это различие не заметно ни пользователям интерфейса Xbase, ни программным приложениям, использующим этот интерфейс. Однако метод передачи вручную, описанный в разделе "Передача на выполнение предложений SQL вручную" на странице 13-4 не поддерживается интерфейсом базы данных Xbase.

Xbase не поддерживает предложение ORDER BY и функции агрегирования типа COUNT, AVG и SUM.

## Примечания к RIS

Этот раздел относится к использованию интерфейса баз данных RIS.

## Столбец nextocc MSCATALOG

Столбец nextocc в MSCATALOG распознает только интерфейс базы данных RIS.

Столбец nextocc содержит очередной доступный номер mslink для таблицы, представленной специфической строкой в MSCATALOG. Если столбец nextocc существует, то MicroStation использует его для определения максимального значения mslink (nextocc-1) и назначения новых номеров mslink. Если столбец nextocc существует, то он должен использоваться для всех таблиц в MSCATALOG; он не может быть пустым для любой строки.

Столбец nextocc имеет то преимущество, что новый номер mslink может быть назначен следующим образом:

1. Получить значение nextocc для специфической таблицы, вызвать его mytabie и сохранить его в переменной currOcc.
2. Передать на выполнение следующее SQL предложение:

```
UPDATE MSCATALOG SET nextocc = currOcc+ 1 WHERE  
tablename = 'mytable' AND nextocc = currOcc
```

Если другое приложение обновило MSCATALOG между шагами 1 и 2 и при этом лишило корректности currOcc, то повторить шаг 2currOcc= currOcc+ 1. После успешного завершения

шага 2 значение currOcc может быть использовано для добавления строки к mytable

Этот подход устраняет необходимость блокирования таблицы, дает максимальное значение mslink, вставляет запись и освобождает таблицу (следовательно завершает транзакцию, в которой была осуществлена блокировка) Этот подход может также быть быстрее, чем операции определения максимума (mslink) для некоторых реляционных баз данных, хотя с соответствующим индексированием большинство баз данных выполняет такие операции довольно быстро

Если в MSCATALOG имеется столбец nextocc, то интерфейс RIS предполагает, что его значение имеет силу. Никакая проверка правильности не выполняется. Необходимо инициализировать столбец nextocc

Использование nextocc может быть опасно для некоторых пользователей. Этот подход предполагает, что каждый пользователь, добавляющий запись в таблицу, очищает номер mslink, несмотря на столбец nextocc в MSCATALOG. Если сетевое приложение добавляет строки к таблице без использования столбца nextocc в MSCATALOG, то MicroStation может позже попытаться вставить строку в таблицу, предполагая, что значение в nextocc не использовалось. Столбцы mslink всегда должны уникально индексироваться, поэтому эта операция скорее не будет выполнена, чем будут продублированы номера mslink. Если по некоторым причинам столбец mslink не имеет уникальной индексации, то возможно дублирование номеров mslink.

## Формирование запросов и сопровождение базы данных

В этой главе описаны процедуры формирования запросов к базе данных, просмотра атрибутов базы данных, удаления подключений, иначе говоря сопровождение базы данных.

### Просмотр атрибутов

Просмотр атрибутов элементов базы данных - это краеугольный камень интерфейса базы данных. Можно графически запрашивать базу данных для просмотра атрибутов, используя средство *Просмотр атрибутов элемента в базе Банных* (см. страницу 17-4). Атрибуты можно просматривать в диалоговом окне Окно SQL или в экранных формах, созданных программным обеспечением базы данных.

### Установка критериев выбора атрибутов для просмотра

Критерии выбора атрибутов для просмотра могут быть установлены и сохранены в столбце sqlreview соответствующей строки в MSCATALOG для желаемой таблицы. Значение sqlreview применяется только к данным, извлеченным для отображения в окне Окно SQL.

**Чтобы проверить соответствие критериям выбора для всех таблиц (значения sql review для всех таблиц, перечисленных в MSCATALOG):**

1. В окне ввода с клавиатуры ввести:

ACTIVE REVIEW \$

или

RA=\$

Значение sql review для первой таблицы выводится в окне Окно SQL.

2. Щелкнуть кнопку Следующая, чтобы просмотреть значения sql review для последующих таблиц.

Чтобы изменить критерии выбора:

1. В окне ввода с клавиатуры ввести:

ACTIVE REVIEW <предложение\_SQL\_SELECT>

или

RA=<предложение\_SQL\_SELECT>

Например, чтобы просмотреть все столбцы в таблице, следует ввести:

RA=SELECT\* FROM <имя\_таблицы>

Это действует по умолчанию для любого столбца sql review со значением NULL.

Чтобы просмотреть заданные столбцы, ввести с клавиатуры:

RA=SELECT<столбец1>, <столбец2>,...  
FROM <имя\_таблицы>

Критерии выбора можно модифицировать непосредственно, передавая на выполнение SQL предложения UPDATE (через диалоговое окно Окно SQL) или используя Диалоговое окно MSCATALOG (see Chapter 17, "Описание интерфейса базы данных," in the on-line *Reference Guide*). Передача на выполнение SQL предложения UPDATE для этой цели может привести к путанице, потому что это предложение должно включать предложение SQL SELECT, как вложенную строку.

## Экранные формы

Очень мощная отличительная черта интерфейса базы данных это способность использовать экранные формы, созданные программой экранных форм этой базы данных, для просмотра и редактирования атрибутов базы данных одновременно. Имя экранной формы, связанной с конкретной таблицей, сохраняется в таблице форм, идентифицированной в необязательном столбце formtable таблицы MSCATALOG или по умолчанию в столбце screenform MSCATALOG.

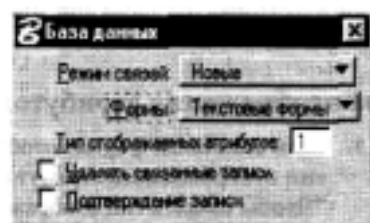
> • Чтобы просмотреть атрибуты базы данных в формате, определенном какой-либо экранной формой:

1. Из подменю База данных меню Установки выбрать пункт Диалог.

Откроется установочное окно База данных.







2. Из меню опций Формы выбрать Текстовые формы.
3. Использовать средство *Просмотр атрибутов элемента в базе данных* для передачи на выполнение данного запроса.

Откроется соответствующее окно и запускается программа экранной формы. Атрибуты базы данных элемента имеют формат, соответствующий определению в экранной форме.

## Отображение атрибутов базы данных в проекте

Можно разместить копию некоторых или всех атрибутов базы данных какого-либо элемента как текст в проекте. Процесс отображения атрибутов очень прост: размещается пустой текстовый узел, который затем обозначается как текстовый узел отображаемых атрибутов. Выбранные атрибуты загружаются в этот текстовый узел отображаемых атрибутов, в соответствии со спецификацией экранной формы или предложением SQL SELECT. Если изменяется информация в базе данных или же Вы изменяете проект экранной формы (или столбцы, выбранные предложением SELECT), то можно обновить проект, просто перезагружая соответствующие текстовые узлы.

Различные типы текстовых узлов отображаемых атрибутов могут быть определены для отображения различных комбинаций атрибутов в различных форматах. Таблица отображаемых атрибутов определяет формат отображения для каждого типа текстового узла отображаемых атрибутов.

Если меню опций Формы в установочном окне База данных установлен в Текстовые формы, то экранная форма,

определенная для типа отображаемого атрибута в столбце formname, используется как шаблон для информации атрибута. Если меню опций Формы установлен в Отсутствуют или нет экранной формы, определенной в столбце form, то предложение SQL SELECT для типа отображаемого атрибута, определенного в данном столбце, задает список загружаемых столбцов. Если нет определенных предложений SELECT, то используется SELECT\*.

## Общая процедура отображения атрибутов



1. Если желаемый формат отображения еще не определен как тип отображаемого атрибута, то создать этот тип. Смотри "Чтобы создать тип отображаемого атрибута:" на странице 14-4.

2. Использовать инструментальное средство *Размещение текстового узла* для размещения текстовых узлов, как свободных знакомест для отображаемых атрибутов.
3. Использовать инструментальное средство *Подключение отображаемых атрибутов* (см. страницу 17-5), чтобы обозначить текстовые узлы, как текстовые узлы отображаемых атрибутов желаемого типа.
4. Использовать инструментальное средство *Подключение отображаемых атрибутов* (см. страницу 17-5), чтобы загрузить атрибуты для отображения в текстовых узлах отображаемых атрибутов.
5. (Необязательно) Если была изменена спецификация типа отображаемого атрибута или значения отображаемых атрибутов и необходимо отразить изменения в проекте, то перезагрузите атрибуты, используя инструментальное средство *Загрузка отображаемых атрибутов* (см. страницу 17-6).

Чтобы создать тип отображаемого атрибута:

1. В диалоговом окне *Окно SQL* передать на выполнение:

```
INSERT INTO <таблица_отображаемых_атрибутов> VALUES
(<номер_типа>,'<имя_экранной_формы>',
'<предложение_SQL_SELECT>')
```

Если определяется предложение SQL SELECT, то имя экранной формы может быть опущено и наоборот.

Чтобы модифицировать тип отображаемого атрибута:

1. В диалоговом окне *Окно SQL* ввести:

```
UPDATE <таблица_отображаемых_атрибутов> SET SQL das =
'<предложение_SQL_SELECT>' WHERE dastype =
<номер_типа>
```

или

```
UPDATE <таблица_отображаемых_атрибутов> SET formname
= '<новое_имя_формы>' WHERE dastype = <номер_типа>
```

Чтобы изменить вид (в противоположность расположению) атрибутов в текстовом узле отображаемых атрибутов, следует использовать установочное окно *Текст* (меню *Элементы/Текст*) и инструментальное средство *Изменение атрибутов текста*.

## Отключение связей с базой данных



Средство *Отключение связи с базой данных* (см. страницу 17-5)

используется для отключения связи от элементов или группы элементов, объединенных в выделенной области. Удаляются связи типа (ов), специфицированного в определении переменной конфигурации MS\_LINKTYPE. Связи других типов и данные

приложений пользователя не нарушаются. Переключатель Удалять связанные записи в установочном окне База данных определяет, будет ли удаляться строка (и), подключенная к конкретному элементу, из таблицы.

Чтобы предотвратить случайные удаления элементов, подключенных к информации базы данных, надо включить предпочтение пользователя Запрет восстановления элементов (см. главу 17, "Описание интерфейса базы данных", *Справочного руководства*) в категории База данных диалогового окна Предпочтения.

## Отключение связей от фрагментов

Чтобы отключить связи с базой данных от компонента какого-либо фрагмента, необходимо вначале разделить фрагмент, используя соответствующее средство, описанное в разделе Инструментальная панель Разделение элементов (см. главу 6 *Руководства пользователя*).

Установочное окно Связи фрагмента с БД используется для связывания библиотеки фрагментов с существующей таблицей

базы данных, подключая связи базы данных к фрагментам библиотеки. Можно также использовать это установочное окно, чтобы удалить связи фрагментов библиотеки от базы данных. Для получения дополнительной информации об установочном окне Связи фрагментов с БД смотри "Утилита Связи фрагментов с БД" в главе 17, "Описание интерфейса базы данных", *Справочного руководства*.

## Проверка связей

При удалении элемента, имеющего подключенную связь с базой данных, MicroStation автоматически удаляет связанную с ним строку из таблицы базы данных, если включен переключатель Удалять связанные записи (см. главу 17, "Описание интерфейса базы данных", *Справочного руководства*) в установочном окне База данных. Однако будет неясная ситуация в противоположном случае. При удалении информации из базы данных без удаления информации связи от элемента, создается связь, называемая **связь сирота**. Связи сироты - это связи без ассоциированных с элементом строк в базе данных.

Быстрая визуальная проверка элементов, имеющих связь с базой данных может быть выполнена с использованием **фильтра выделенной области**, запроса, определяющего какие элементы со связями к строкам в таблице выбираются для операций с выделенной областью. Эта процедура не идентифицирует связи сироты, но показывает элементы, не имеющие никаких связей. Например, если каждый стол в помещении офиса имеет какую-либо связь с базой данных, то все они будут подсвечены. Столы, у которых отсутствуют связи, не будут подсвечены.

Для ознакомления с более полной проверкой следует прочитать раздел Использование утилиты для проверки базы данных (см. страницу 14-7). Она генерирует список всех элементов в файле проекта со связями выбранного типа с базой данных и идентифицирует связи сироты.

Чтобы визуально проверить связи базы данных:



1. В окне ввода команд ввести:

DEFINE SEARCH SELECT\* FROM <имя\_таблицы>

или

DS=SELECT\* from <имя\_таблицы>

Это устанавливает фильтр выделенной области для всех строк в базе данных, которые имеют mslink подключение.

Сообщение "Фильтр выделенной области: вкл." выводится в строке состояния.

Использовать средство *Определение выделенной области* для заключения в выделенную область части проекта, представляющей интерес (или всего проекта).

2. В окне ввода команд ввести:

UC=hilite

Все элементы со связями с базой данных подсвечиваются.

Команда UC= активизирует команду пользователя (макрос) "hilite.ucm". Для получения дополнительной информации о командах пользователя, смотри "Команды пользователя" в главе 8 *Руководства пользователя*.

## Использование утилиты для проверки базы данных



Утилита для проверки базы данных ("dbcheck-ma") автоматизирует обнаружение и исправление связей сирот.

**Чтобы загрузить утилиту для проверки базы данных и открыть установочное окно:**

1. В установочном блоке MDL (меню Утилиты/MDL

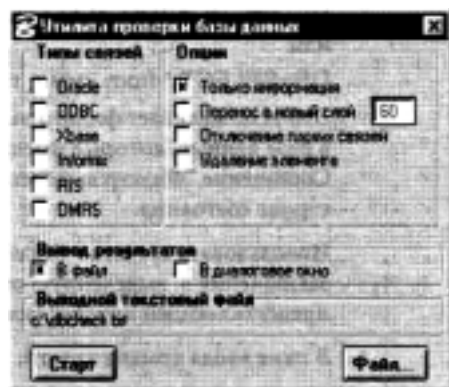
приложения) выбрать DBCHECK в поле списка Доступные приложения.

2. Щелкнуть кнопку Загрузить.

или

1. В окне ввода с клавиатуры ввести:

MDL LOAD DBCHECK



Чтобы вновь открыть установочное окно утилиты проверки базы данных при загруженной утилите:

1. В окне ввода с клавиатуры ввести:

DVCSHECK

Чтобы найти связи сироты данного типа (ов):

1. В секции Типы связей установочного окна Утилита проверки базы данных, выбрать тип (ы) связи для анализа.

2. В секции Опции включить переключатель Только информация.

Сначала следует всегда выполнить данную утилиту с включенным переключателем Только информация перед использованием ее с другими опциями, которые модифицируют файл проекта.

3. Чтобы определить, что информация связи будет записываться в файл, идентифицированный в поле Выходной текстовый файл, надо включить переключатель В файл. и/или

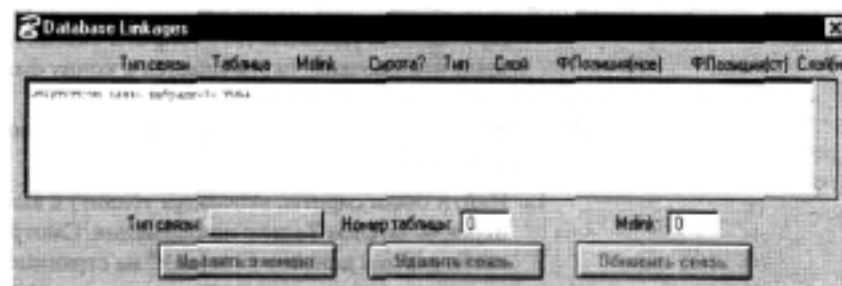
Чтобы определить, что информация связи будет выводиться в установочное окно Database Linkages, надо включить переключатель В диалоговое окно.

4. (Необязательно) Чтобы изменить имя текстового файл для вывода вместо файла по умолчанию, "dbcheck-txt", надо щелкнуть кнопку Файл... и определить файл, используя диалоговое окно File Name. Затем щелкнуть кнопку ОК, чтобы вернуться в установочное окно описываемой утилиты.

5. Щелкнуть кнопку Старт.

Утилита генерирует определенный текстовый файл и/или открывает установочное окно Database Linkages. В любом случае информация связи форматируется как таблица с одной связью на строку, отсортированной по типу, затем по номеру объекта и затем по значению ключа mslink. Связи сироты идентифицируются буквой "Y" в столбце Сирота?.

Установочное  
окно Database  
Linkages



Манипулирование элементами со связями сиротами

Желательно выполнить какие-либо из следующих корректирующих действий после обнаружения связей сирот:

- Изолировать элементы со связями сиротами на определенном слое.
- Удалить связи сироты.
- Удалить элементы, к которым подключены связи сироты.

| Установочное окно Database Linkages позволяет манипулировать отдельными связями. См. "Манипулирование отдельными связями" на странице 14-10.

Чтобы переместить все элементы со связями сиротами данного типа (ов) на один слой:

1. Найти связи сироты, используя утилиту с включенным переключателем Только информация. См. "Чтобы найти связи сироты данного типа (ов):" на странице 14-8.
2. В секции Опции установочного окна данной утилиты включить переключатель Перенос в новый слой и, если требуется, ввести с клавиатуры значение в поле справа для замены цифры 60 как номера слоя по умолчанию.
3. Щелкнуть кнопку Выполнить.

Чтобы удалить все связи сироты данного типа (ов):

1. Найти связи сироты, используя утилиту с включенным переключателем Только информация. См. "Чтобы найти связи сироты данного типа (ов):" на странице 14-8.
2. В секции Опции установочного окна данной утилиты включить переключатель Отключение плохих связей.
3. Щелкнуть кнопку Старт.

При каждом удалении связи сироты элемент, к которому она была подключена, перемещается к концу файла проекта.

Чтобы удалить элементы, к которым подключены связи сироты данного типа (ов):

1. Найти связи сироты, используя утилиту с включенным переключателем Только информация. См. "Чтобы найти связи сироты данного типа (ов):" на странице 14-8.
2. В секции Опции установочного окна данной утилиты включить переключатель Удаление элемента.
3. Щелкнуть кнопку Старт.

### **Манипулирование отдельными связями**

В дополнение к идентификации связей сирот, установочное окно Database Linkages позволяет манипулировать отдельными связями и элементами, к которым подключены эти связи.

#### **• Чтобы изменить тип связи:**

1. В поле списка выбрать связь.
2. Из меню опций Тип связи выбрать желаемый тип.
3. Щелкнуть кнопку Обновить связь.

<sup>D</sup> Изменение DMRS связи: Нельзя изменить ни DMRS связь на другой тип связи, ни изменить другие типы связи на DMRS. Все другие типы связи могут быть изменены.

#### **- Чтобы изменить номер объекта в какой-либо связи:**

1. В поле списка выбрать связь.
2. В поле Номер таблицы ввести с клавиатуры желаемый номер.

3. Щелкнуть кнопку Обновить связь.

Чтобы изменить значение ключа mslink в какой-либо связи:

1. В поле списка выбрать связь.

2. В поле Mslink ввести с клавиатуры желаемое значение.

3. Щелкнуть кнопку Обновить связь.

При нажатии кнопки Обновить связь и выбранной связи сироты, символ "Y" в строке поля списка для данной связи изменяется на

"N/A"

Чтобы удалить элемент с какой-либо связью с базой данных:

1. В поле списка выбрать любую связь с элементом.

2. Щелкнуть кнопку Удалить элемент.

Все связи элемента становятся тусклыми в поле списка, чтобы показать, что данный элемент удален.

Чтобы удалить какую-либо связь от элемента:

1. В поле списка выбрать связь.

2. Щелкнуть кнопку Удалить связь.

## Примечания к Xbase

Если включен переключатель Удалять связанные записи (см. главу 17, "Описание интерфейса базы данных", *Справочного руководства*) в установочном окне База данных, то MicroStation автоматически удалит строку из таблицы при удалении элемента, с которым была связана эта строка. В базах данных Xbase строка физически не удаляется из таблицы, пока не выполняется выход из системы базы данных и пока не выполняется оператор PASC. Необходимо также переопределить индексы после упаковки строк.

Независимо от используемого приложения при работе с базой данных, в некоторый момент возникает необходимость в генерации отчетов для передачи другим людям определенной информации о проекте.

Отчет может быть очень простым, как, например, список деталей для механического узла, или сложным, как, например, эксплуатационный план для всего оборудования в выбранной зоне большого предприятия, у которого еще не кончился гарантийный срок и требуется остановка оборудования для ремонта.

Независимо от приложения, главным достоинством интерфейса с базой данных является способность объединить возможности запросов к базе данных с графическим изображением для анализа, манипулирования и подготовки отчетов.

В этой главе описываются процедуры генерации отчетов, основанных на атрибутах базы данных элементов. Кроме того, эта глава содержит краткий обзор методов форматирования отчета и возможностей вывода информации из баз данных, поддерживаемых в MicroStation. Упражнения по созданию отчетов смотри в разделе "Создание связей с базой данных" в главе 14 *Учебного пособия*.

## Таблицы отчетов



Отчеты по базе данных в MicroStation могут отображаться в виде чертежей проекта с выделенной информацией отчета или в виде таблиц отчетов.

Инструментальное средство *Генерация таблицы отчета* используется для генерации таблиц отчетов для каждой таблицы, содержащей строки, подключенные к элементам, содержащимся в выделенной области. **Таблица отчетов** содержит строку для каждой записи выбранной таблицы, подключенной к элементу в выделенной области. Для форматирования и печати таблиц отчетов требуется программное обеспечение вывода отчетов, которое обычно входит в состав используемого пакета реляционной базы данных (смотри "Форматирование отчетов и их вывод" на странице 15-5).

Так как для каждого появления связанного элемента записывается некоторая строка в таблицу отчетов, таблица дает точный подсчет связей, которые были подключены при установленном меню опций Режим связей Дублирующие. Многократное использование режима связей Дублирующие позволяет подключить к одной строке базы данных некоторое множество элементов. В одной таблице отчетов могут быть смешаны связи, выполненные в различных режимах связей. Таким образом, можно получить сотни идентичных элементов в проекте, связанных с одной и той же строкой, и тем не менее сгенерировать таблицу отчетов для подготовки спецификаций и количественного контроля состава элементов.

Необходимо также отметить, что связи, созданные при установленном меню опций Режим связей - Информационные, не включаются в таблицы отчетов.

## Общая процедура генерации таблиц отчетов

1. Проверить, что в таблице MSCATALOG определены имена каждой таблицы, имеющей строки, подключенные к элементам в анализируемой части проекта. Смотри "Именованная таблица" на странице 15-3.
  2. (Необязательно) Чтобы выбрать критерии для подготовки отчета, основанные на атрибутах, в дополнение к географическим критериям, определенным выделенной областью, необходимо определить фильтр (ы) выделенной области (DS=). Смотри "Фильтры выделенной области" на странице 15-4.
  3. Разместить выделенную область, так чтобы она включала в себя анализируемую часть проекта. Для этого следует использовать средство *Определение выделенной области*.
  4. Использовать инструментальное средство *Генерация таблицы отчета* (см. страницу 17-7).
- Первые три шага могут быть выполнены в любом порядке.



Каждый раз при генерации таблицы отчетов любая существующая таблица отчетов с тем же самым именем заменяется новой.

## Именованние таблиц

Перед генерацией таблицы отчетов ее имя должно быть определено в столбце reporttable таблицы MSCATALOG.

Чтобы проверить имена таблиц отчетов для всех таблиц, перечисленных в MSCATALOG:

1. В окне команд ввести с клавиатуры:

ACTIVE REPORT \$

или

RS=\$

Значение столбца reporttable для первой таблицы выводится в диалоговом окне Окно SQL.

Щелкнуть кнопку Следующая, чтобы просмотреть значения reporttable для следующих таблиц.

**Чтобы изменить имя таблицы отчетов:**

1. В окне ввода команд ввести с клавиатуры:

ACTIVE REPORT

<имя\_таблицы>: <имя\_таблицы\_отчетов>

или

P8=<имя\_таблицы>: <имя\_таблицы\_отчетов>

Например, чтобы назвать таблицу отчетов parts\_report, создаваемую для таблицы parts, ввести с клавиатуры:

RS=parts: parts\_report

Диалоговое окно MSCATALOG (см. главу 17, "Описание интерфейса базы данных", *Справочного руководства*) может также использоваться для просмотра и изменения спецификации имени

таблицы отчетов. Следует использовать ввод в командной строке MicroStation команды Set database..

## Фильтры выделенной области

Фильтры выделенной области используются вместе с инструментальным средством *Генерация таблицы отчета.*, а также для генерации отчетов в форме выводимых на плоттер проектов. **Фильтр выделенной области** определяет неграфические критерии выбора элементов, используемых для операций с выделенной областью. Таблицы отчетов, созданные с использованием фильтров выделенной области, включают строки только для элементов, содержащихся в выделенной области и удовлетворяющих заданным критериям. Это дает дополнительное преимущество, позволяя графически оперировать с выбранными элементами.

Строка fencefilter в MSCATALOG содержит спецификацию фильтра выделенной области. Каждая таблица, внесенная в список в MSCATALOG, может иметь уникальный фильтр выделенной области.

**Чтобы отказаться от определенных спецификаций фильтра выделенной области и отключить выбор с использованием фильтра выделенной области:**

1. В окне команд ввести с клавиатуры:

DEFINE SEARCH none

или

DS=none

Значения столбца fencefilter удаляются из каждой строки в MSCATALOG.

**Чтобы определить фильтр выделенной области и активизировать выбор с использованием фильтра выделенной области:**

1. В окне команд ввести с клавиатуры:

DEFINE SEARCH <предложение\_SQL\_SELECT>

или

DS=<предложение\_SQL\_SELECT>

Определенное предложение SELECT сохраняется в столбце fencefilter строки для определенной таблицы в MSCATALOG.

Фильтр выделенной области может также быть определен непосредственной передачей на выполнение предложения SQL UPDATE (в диалоговом окне Окно SQL) или используя Диалоговое окно MSCATALOG (см. главу 17, "Описание интерфейса базы данных". *Справочного руководства*). Выполнение предложения SQL UPDATE для этой цели может быть безуспешным, если это предложение должно быть объединено с предложением SQL SELECT в одной строке.

Можно использовать макрос команд пользователя "hilite.bas" для подсвечивания каждого элемента в выделенной области, удовлетворяющего критериям фильтра. Затем, используя средство *Изменение атрибутов элемента* при включенном параметре ВО (Выделенная область), можно изменить слой, цвет, стиль или толщину линий подсвеченных элементов. Эти изменения могут быть отражены в чертеже для показа результатов анализа. (Дополнительную информацию об активизации команд пользователя смотри в разделе "Команды пользователя" в главе 8 *Руководства пользователя*.)

## Форматирование отчетов и их вывод

Таблица отчетов, созданная из MicroStation, может быть отформатирована для вывода как отчета, используя программу оформления отчетов базы данных. Ниже приводится краткое описание программ оформления отчетов, доступных для баз данных, поддерживаемых в MicroStation.

### Oracle

Oracle поддерживает оформление отчетов как в интерфейсе SQL\* Plus так и в мощном пакете оформления сложных отчетов SQL\* ReportWriter. SQL\* Plus поддерживает большинство функций оформления отчетов. Он позволяет управлять размещением столбцов, размерами страницы, добавлять заголовки и нижние колонтитулы и управлять разрывами для промежуточных итогов. Можно также объединять таблицы и создавать итоговые отчеты.

SQL\* ReportWriter позволяет создавать отчеты для одиночных и множественных запросов, используя ряд сложных форматов отчета. Он использует подход заполнения по образцу, так что не требуется какое-либо программирование. Однако, обширность

возможностей может отпугнуть пользователей, предъявляющих простые требования по оформлению отчетов.

## **INFORMIX**

Программа оформления отчетов, поставляемая в составе INFORMIX - ACE. Она позволяет определять и компилировать отчеты. Отчеты могут быть любыми - от простых распечаток до сложных многотабличных отчетов, требующих подведения окончательных и промежуточных итогов. Оформление отчета может быть запущено изнутри INFORMIX-SQL или из системной командной строки.

Программа оформления отчетов ACE функционально полна и более чем достаточна для удовлетворения потребностей типичного пользователя интерфейса INFORMIX MicroStation.

## **Xbase**

Семейство программ Xbase использует общий формат файла базы данных. Следовательно будет пригодна любая программа оформления отчетов, поддерживающая этот формат. Такие программы включены в состав dBASE и FoxPro.

Кроме того сам язык команд Xbase непосредственно поддерживает простую интерактивную генерацию отчетов. Можно использовать команды SET ALTERNATE TO и SET ALTERNATE ON, наряду с командой LIST, для сохранения отформатированного изображения, как текстового файла. Этот файл может быть открыт любым текстовым процессором для заключительного форматирования.

Следует обратить внимание, что таблицы отчетов Xbase имеют расширение ".prt", но они все еще используют формат файла "dbf". Необходимо только включить расширение ".prt" в команду dBASE USE.

## **Программные приложения баз данных**

Язык разработки приложений MicroStation (MDL) может использоваться для расширения возможностей интерфейса базы данных и адаптации ее для различных применений. Подобно другим MDL приложениям, расширяющим функциональные возможности MicroStation, MDL приложения для баз данных могут быть приобретены у разработчиков программного обеспечения или разработаны с использованием MDE (Среда разработки MicroStation) - инструментария, поставляемого с MicroStation.

В этой главе описаны некоторые разновидности доступных MDL приложений для баз данных. Упражнения, связанные с использованием MDL приложений, расширяющих возможности интерфейса базы данных, смотри в разделе "Создание связей с базой данных" в главе 14 *Учебного пособия*.

### **Доступные программные приложения баз данных**

Ниже перечислены разновидности коммерчески доступных MDL приложений для баз данных, иллюстрирующие как используемые

приложения позволяют расширить возможности интерфейса базы данных:

- Архитектурные приложения для автоматической генерации спецификаций, ведомостей, планов, распечаток проектов и окружающего рельефа.
- Приложения для объединения кадастровой информации и других атрибутов с выводимыми координатными данными и геометрической информацией на планах местности.
- Приложения для визуализации использования окружающей среды.
- Приложения для преобразования географических данных в различные системы координат.
- Приложения для непосредственной модификации строк базы данных, используя диалоговые окна.
- Машиностроительные приложения проекта для автоматической генерации перечней деталей и номеров позиций и детализовки сборочных чертежей.

Общая информация о приобретении и использовании MDL приложений приведена в разделе "MDL приложения" в главе 9 *Руководства пользователя*.

## Разработка MDL приложений для баз данных

MDL (язык разработки приложений MicroStation) - основной программный интерфейс для разработки приложений MicroStation. MDL - полная реализация стандарта ANSI для языка программирования Си. MDL обеспечивает разработчикам доступ ко всем графическим функциям и функциям интерфейса баз данных MicroStation, а также полный контроль над графическим интерфейсом пользователя, использующим меню и диалоговые окна.

Хотя основные правила программирования MDL не относятся к содержанию этого руководства, пример приложения "gis.ma" позволяет оценить потенциал MDL при разработке заказных приложений для баз данных. Общую информацию об MDL и

разработке приложений на MDL смотри в разделе "MDL как альтернатива" в главе 6 *Руководства по MicroStation BASIC*.

## Инструментальная панель База данных

Инструментальные средства в инструментальной панели База данных используются для установления, просмотра и удаления подключений базы данных, генерации отчетов и отображения атрибутов базы данных в текстовых узлах отображаемых атрибутов. Инструментальная панель База данных открывается при выборе пункта База данных из меню Инструменты.

Чтобы:	Выбрать в инструментальной панели База данных:
Подключить активный объект к элементу.	 Подключение активного объекта (страница 17-2)
Отобразить режим подключения.	 Показ режима связей (страница 17-2)
Отобразить активный объект.	 Отображение активного объекта (страница 17-3)
Установить строку базы данных как активный объект.	 Графическое определение активного объекта (страница 17-3)
В интерактивном режиме отобразить атрибуты элемента базы данных.	 Просмотр атрибутов элемента в базе данных (страница 17-4)
Отключить строки базы данных от элемента.	 Отключение связи с базой данных (страница 17-5)
Подключить отображаемые атрибуты к элементу.	 Подключение отображаемых атрибутов (страница 17-5)
Загрузить текстовые узлы отображаемых атрибутов, содержащиеся в выделенной области, атрибутами данных.	 Загрузка отображаемых атрибутов (страница 17-6)
Генерировать таблицу отчетов для каждой таблицы с подключениями к элементам в выделенной области.	 Генерация таблицы отчета (страница 17-7)

## Подключение активного объекта



Используется для подключения активного объекта к элементу или набору элементов в выделенной области при включенном параметре ВО (выделенная область). Строки таблицы базы данных могут быть подключены к любому простому или сложному элементу, а отдельный элемент может иметь много подключений ко многим таблицам.

Режим связей устанавливается в меню опций Режим связей в установочном окне База данных, которое открывается при выборе пункта База данных> Диалог из меню Установки.

Режим связей:	Активный объект:
Новые	Копируется и добавляется к базе данных как новая строка, а также подключается к идентифицированному элементу.
Любой кроме Новых	Это некоторая строка в базе данных, но не строка прототипа. Эта строка непосредственно подключается к элементу без создания копии строки.

Чтобы подключить активный объект к какому-либо элементу:

1. Выбрать инструментальное средство *Подключение активного объекта*.

2. Идентифицировать элемент, к которому необходимо подключить активный объект.

3. Принять элемент.

Ввод с клавиатуры: ATTACH AE

## Показ режима связей



Используется для отображения режима связей в окне команд.

**Чтобы отобразить режим связей:**

- Выбрать инструментальное средство *Показ режима связей*.

Ввод с клавиатуры: ACTIVE LINKAGE

## Отображение активного объекта



Используется для отображения активного объекта.

**Чтобы отобразить активный объект:**

1. Выбрать инструментальное средство *Отображение активного объекта*.

Строка базы данных, соответствующая активному объекту, выводится в диалоговом окне *Окно SQL*.

Ввод с клавиатуры: SHOW AE

или

AE=\$

## Графическое определение активного объекта



Используется для определения строки базы данных как активного объекта. Определенный! активный объект может быть подключен к элементу или отредактирован.

**Чтобы установить какую-либо строку базы данных как активный объект:**

1. Выбрать инструментальное средство *Графическое определение активного объекта*.
2. Идентифицировать элемент, к которому подключена строка базы данных.
3. Принять элемент.

Если в установочном окне База данных переключатель Подтверждение записи - выключен, то первая строка, подключенная к элементу, выбирается в качестве активного объекта.

Если переключатель Подтверждение записи - включен, то первая строка выводится в диалоговом окне Окно SQL.

4. Если имеется более одной подключенной строки, то можно перебирать их нажимая на кнопку Следующая. Чтобы принять строку как активный объект, следует щелкнуть кнопку Принять. Чтобы вернуться на шаг 2, необходимо щелкнуть кнопку Прервать.

Ввод с клавиатуры: DEFINE AE

## Просмотр атрибутов элемента в базе данных



Используется для отображения в интерактивном режиме атрибутов элемента в базе данных.

- Если меню опций Формы установлено в состояние Текстовые формы, то для отображения каждой подключенной строки используется экранная форма, определенная в столбце screen-form таблицы MSCATALOG. Экранные формы должны находиться в каталоге, определяемом переменной конфигурации MS\_DBASE.
- Если меню опций Формы установлено в состояние Отсутствуют или в столбце screenform не определена какая-либо экранная форма, то формат отображения определяется предложением SQL SELECT команды ACTIVE REVIEW (RA=). Если нет определенных предложений SELECT, то используется SELECT\*. Если имеется несколько строк, подключенных к элементу, то они отображаются последовательно. Строка (и) выводятся в диалоговом окне Окно SQL.

Меню опций Формы устанавливается в установочном окне База данных, которое открывается при выборе пункта База данных> Диалог из меню Установки или командой с клавиатуры - FORMS.

### Чтобы просмотреть атрибуты элемента в базе данных:

1. Выбрать инструментальное средство *Просмотр атрибутов элемента в базе данных*.
2. Идентифицировать элемент, атрибуты которого в базе данных необходимо отобразить.
3. Принять элемент.

Отображаются выбранные столбцы первой подключенной строки. Если имеется более одной подключенной строки, можно переходить к следующей строке, выходя из пакета

программ для работы с экранными формами или щелкая кнопку Следующая в диалоговом окне Окно SQL. Чтобы прервать отображение строк и вернуться на шаг 2, необходимо нажать кнопку Сброс (программа для работы с экранной формой) или Прервать (диалоговое окно Окно SQL).

Ввод с клавиатуры: REVIEW

## Отключение связи с базой данных



Используется для отключения строк базы данных от элемента или набора выбранных элементов в выделенной области при включенном параметре ВО (выделенная область). Строки удаляются из базы данных, если включен переключатель Удалять связанные записи в установочном окне База данных.

Это инструментальное средство может использоваться для отключения связей только того типа (ов), который описан в переменной конфигурации MS\_LINKTYPE.

### Чтобы отключить строки базы данных от элемента:

1. Выбрать инструментальное средство *Отключение связи с базой данных*.
2. Идентифицировать элемент.
3. Принять элемент.

Ввод с клавиатуры: DETACH

## Подключение отображаемых атрибутов



Используется для назначения пустого текстового узла как текстового узла отображаемых атрибутов, который может быть выбран для отображения атрибутов с помощью инструментального средства *Загрузка отображаемых атрибутов*.

Некоторая специальная связь с базой данных подключается к этому текстовому узлу: эта связь с базой данных относится к типу Информационные и не может участвовать в генерации отчетов.

### - Чтобы назначить текстовый узел отображаемых атрибутов:

1. Выбрать инструментальное средство *Подключение отображаемых атрибутов*.
2. Идентифицировать элемент, к которому подключена требуемая строка базы данных.
3. Принять элемент.



Если переключатель Подтверждение записи в установочном окне База данных - включен, то первая строка выводится в диалоговом окне Окно SQL. Если имеется более одной подключенной строки, то можно перебирать их, нажимая на кнопку Следующая. Чтобы принять строку как строку, подключаемую к данному текстовому узлу, необходимо щелкнуть кнопку Принять или нажать <Ctrl-A>. Чтобы вернуться на шаг 2, надо щелкнуть кнопку Прервать.

#### 4. Идентифицировать пустой текстовый узел.

Если переключатель Подтверждение записи - выключен, то первая строка, подключенная к элементу, подключается также и к текстовому узлу.

Ввод с клавиатуры: ATTACH DA

## Загрузка отображаемых атрибутов



Используется для загрузки атрибутов базы данных в текстовый узел отображаемых атрибутов. Если база данных была изменена, то это инструментальное средство можно использовать для обновления текста в текстовом узле отображаемых атрибутов. Это средство имеет переключатель ВО (выделенная область) для операций с выделенной областью.

- Если меню опций Формы установлено в состояние Текстовые формы, то для обеспечения шаблона информации атрибута используется экранная форма, определенная в столбце form таблицы отображаемых атрибутов. Экранные формы должны находиться в каталоге, определяемом переменной конфигурации MS\_DBASE.
- Предложение SQL SELECT для таблицы отображаемых атрибутов, определенное в столбце sqldas этой таблицы, определяет список загружаемых столбцов. Предполагается, что переключатель Формы установлен в состояние Отсутствуют или в столбце form не определена какая-либо экранная форма. Если нет определенных предложений SELECT, то используется SELECT\*.

#### **Чтобы загрузить текстовый узел отображаемых атрибутов:**

1. Выбрать инструментальное средство *Загрузка отображаемых атрибутов*.
2. Идентифицировать текстовый узел.
3. Принять текстовый узел.

Ввод с клавиатуры: LOAD DA

Многие шрифты MicroStation являются пропорциональными, это означает, что столбцы в экранной форме будут выстраиваться неправильно при преобразовании формы в текстовые элементы в файле проекта. По этой причине рекомендуется использование

шрифта с фиксированной шириной символа (например, шрифт 3) для сохранения интервала между столбцами.

## Генерация таблицы отчета



Используется для генерации таблицы отчетов для любой таблицы, имеющей строки, подключенные к элементам в выделенной области. Отдельная строка записывается в таблице отчетов для каждой подключенной строки. Имя каждой таблицы отчетов определяется командой ACTIVE REPORT (RS=).

### **Структура каждой таблицы отчетов идентична структуре основной таблицы:**

1. Используя инструментальное средство *Определение выделенной области*, определить требуемую область.
2. Выбрать инструментальное средство *Генерация таблицы отчета*.
3. Ввести информационную точку.

Каждая строка, подключенная к элементу в выделенной области, копируется в таблицу отчетов, соответствующую основной таблице, содержащей эту строку. Таблица отчетов содержит одну строку для каждой связи с базой данных, а строки могут дублироваться.

После генерации таблицы отчетов заключительное форматирование отчета выполняется программой оформления отчетов конкретной базы данных.

Ввод с клавиатуры: FENCE REPORT

При записи сгенерированной таблицы отчетов она

- перезаписывается поверх любых существующих таблиц с тем же именем.

| Содержимое выделенной области определяется режимом выбора выделенной области.