Oracle® Migration Workbench

Reference Guide for MySQL 3.22, 3.23 Migrations

Release 9.2.0 for Microsoft Windows 98/2000, Microsoft Windows NT and Red Hat Linux 6.2

March 2002

Part Number: A97249-01

This reference guide describes how to migrate from MySQL to an Oracle9*i* or Oracle8*i* database.

Oracle Migration Workbench Reference Guide for MySQL 3.22, 3.23 Migrations, Release 9.2.0 for Microsoft Windows 98/2000, Microsoft Windows NT and Red Hat Linux 6.2

Part Number: A97249-01

Contents

3 MySQL Data Types, Reserved Words, and Operators

4 Troubleshooting

A MySQL Error Messages

Index

Send Us Your Comments

Oracle Migration Workbench Reference Guide for MySQL 3.22, 3.23 Migrations, Release 9.2.0 for Microsoft Windows 98/2000, Microsoft Windows NT, and Red Hat Linux 6.2

Part Number: A97249-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Email - infomwb_ww@oracle.com
- FAX - +353-1-803-1899
- Postal service:

  Documentation Manager
  Migration Technology Group
  Oracle Corporation
  Eastpoint Business Park
  Dublin 3
  Ireland

Preface

The Oracle Migration Workbench *Reference Guide for MySQL 3.22, 3.23 Migrations* provides detailed information about migrating a database from MySQL to Oracle9*i*, Oracle8, or Oracle8*i*. It describes several differences between MySQL and Oracle. It also outlines how those differences are dealt with by the Oracle Migration Workbench (Migration Workbench) during the conversion process.

This chapter contains the following sections:

- Audience
- What You Should Already Know
- Structure of this Guide
- Using this Guide
- Documentation Accessibility
- Accessibility of Code Examples in Documentation
- Related Documentation
- Conventions

Audience

This guide is intended for anyone who is involved in converting a MySQL database to Oracle using the Migration Workbench.

What You Should Already Know

You should be familiar with relational database concepts. You should also be familiar with the operating system environments where you are running MySQL and Oracle.

Structure of this Guide

This reference guide is organized as follows:

Chapter 1, "Overview"

Introduces the concepts and features of the Migration Workbench.

Chapter 2, "Migration Process"

Describes the architecture of MySQL and Oracle. It explains how to prepare the MySQL database for migration. This chapater also describes how to migrate from MySQL to Oracle using the Migration Workbench.

Chapter 3, "MySQL Data Types, Reserved Words, and Operators"

Provides the data types from MySQL to Oracle and provides a list of Oracle9*i* and Oracle8*i* reserved words.

Chapter 4, "Troubleshooting"

Describes how to solve problems using the Migration Workbench.

Appendix A, "MySQL Error Messages"

Proviedes a list of MySQL error messages and possible solutions.

Using this Guide

Every reader of this reference guide should read Chapter 1, "Overview". That chapter provides an introduction to the concepts and terminology of the Migration Workbench.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Related Documentation

For more information, see these Oracle Migration Workbench resources:

- Oracle Migration Workbench Frequently Asked Questions (FAQ)
- Oracle Migration Workbench Release Notes
- Oracle Migration Workbench Online Help

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and you can do it at:

http://otn.oracle.com/membership/index.htm

If you already have a user name and password for OTN, then you can go directly to the Migration Workbench documentation section of the OTN Web site at:

http://otn.oracle.com/tech/migration/workbench

Conventions

This section describes the conventions used in the text and code examples of the this documentation. It describes:

- Conventions in Text
- Conventions in Code Examples

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold type indicates GUI options. It also indicates terms that are defined in the text or terms that appear in a glossary, or both.	The C datatypes such as **ub4**, **sword**, or **OCINumber** are valid. When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles, emphasis, syntax clauses, or placeholders.	*Reference Guide* Run `U`*`old_release`*`.SQL` where *old_release* refers to the release you installed prior to upgrading.
UPPERCASE monospace (fixed-width font)	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, user names, and roles.	You can specify this clause only for a `NUMBER` column. You can back up the database using the `BACKUP` command.
lowercase monospace (fixed-width font)	Lowercase monospace typeface indicates executables and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, user names and roles, program units, and parameter values.	Enter `sqlplus` to open SQL*Plus. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
Square Brackets [ ]	Indicates that the enclosed arguments are optional. Do not enter the brackets.	DECIMAL (digits [, precision])
Curly Braces { }	Indicates that one of the enclosed arguments is required. Do not enter the braces.	{ENABLE | DISABLE}
Vertical Line |	Separates alternative items that may be optional or required. Do not type the vertical bar.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]

Ellipses ...	Indicates that the preceding item can be repeated. You can enter an arbitrary number of similaritems. In code fragments, an ellipsis means that code not relevant to the discussion has been omitted. Do not type the ellipsis	CREATE TABLE ... AS subquery;
		SELECT col1, col2, ... , col*n* FROM employees;

Italics	Indicates variables that you must supply particular values.	CONNECT SYSTEM/*system_password*

UPPERCASE	Uppercase text indicates case-insensitive filenames or directory names, commands, command keywords, initializing parameters, data types, table names, or object names. Enter text exactly as spelled; it need not be in uppercase	SELECT last_name, employee_id FROM employees;
		SELECT * FROM USER_TABLES;
		DROP TABLE hr.employees;

lowercase	Lowercase words in example statements indicate words supplied only for the context of the example. For example, lowercase words may indicate the name of a table, column, or file.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr

1

Overview

This chapter introduces the Migration Workbench. It includes information on the following:

- Introduction
- Product Description
- Features
- Glossary

Introduction

MySQL is an open-source relational database management system (RDBMS). It uses a client/server architecture and is a multi-threaded, multi-user database server. It was designed specifically to be a fast server; therefore, it does not provide many of the features provided by other relational database systems, such as subselects, foreign keys, referential integrity, stored procedures, triggers, or views. In addition, it contains a locking mechanism that is not adequate for tables containing many write actions occurring simultaneously from different users. It is also lacking in reference to support for software applications and tools.

The Oracle RDBMS is a modern, scaleable, high performance database server that can run on a wide range of computers from PCs to mainframes. Oracle operates in a network, client/server environment. It can support tens, hundreds, or thousands of users, depending on the server.

This guide explains how to migrate from MySQL to Oracle using the Migration Workbench. It also provides guidelines on how to modify the MySQL applications to work with the new Oracle database. If you have an investment in MySQL applications, you can retain this investment while adding the advanced features of Oracle to the application architecture.

Product Description

The Migration Workbench is a tool that simplifies the migration of data and applications from a MySQL 3.23 environment to an Oracle9*i*, Oracle8*i*, or Oracle8 destination database. The Migration Workbench allows you to migrate an entire application system in an integrated, visual environment. This involves migrating the database schema including default values and indexes.

The Migration Workbench provides an intuitive and informative user interface (UI) and a series of wizards to simplify the migration process. To ensure portability, all components of the Migration Workbench are written in Java.

The Migration Workbench uses a repository to store migration information. This allows you to query the initial state of the application before migration. By initially loading the components of the application system that you can migrate into a repository, you can work independently of the production application. You can use the Migration Workbench to migrate the following:

- Users
- Tables
- Databases
- Primary keys
- Indexes
- Column data

Features

The Migration Workbench is a wizard-driven tool. It is composed of core features and MySQL migration specific features. The Migration Workbench allows you to:

- Migrate a complete MySQL database to an Oracle9*i*, Oracle8*i*, or Oracle8 database.
- Display a representation of the source database and its Oracle equivalent.
- Migrate users, tables, table level privileges, indexes, and primary keys.
- Migrate data types, such as text, date, and ENUM.
- Migrate MySQL databases with MySQL tables.
- Display a representation of the source database and its Oracle equivalent.
- Generate and view a report of the migration.

- Customize users, tables, indexes, and tablespaces.
- Customize the default data type mapping rules.
- Create ANSI-compliant names.
- Resolve conflicts, such as Oracle reserved words, automatically.
- Delete and rename objects in the Oracle Model.
- Migrate individual table data.
- Use the offline data loading capability when migrating the data to Oracle9*i* and Oracle8*i*.

Glossary

The following terms are used to describe the Migration Workbench:

Application System is the database schema and application files that have been developed for a database environment other than Oracle, for example, MySQL.

Capture Wizard is an intuitive wizard that takes a snapshot of the data dictionary of the source database, loads it into the Source Model, and creates the Oracle Model.

Dependency is used to define a relationship between two migration entities. For example, a database view is dependent upon the table it references.

Destination Database is the Oracle database to which the Migration Workbench migrates the data dictionary of the source database.

Migration Component is part of an application system that can be migrated to an Oracle database. Examples of migration components are tables and stored procedures.

Migration Entity is an instance of a migration component. The table EMP would be a migration entity belonging to the table MIGRATION COMPONENT.

Migration Wizard is an intuitive wizard that helps you migrate the source database to Oracle.

Migration Workbench is the graphical tool that allows migration of an application system to an Oracle database environment.

Navigator Pane is the part of the Migration Workbench User Interface that contains the tree views representing the Source Model and the Oracle Model.

Oracle Model is a a series of Oracle tables that is created from the information in the Source Model. It is a visual representation of how the source database will look when generated in an Oracle environment.

Properties Pane is the part of the Migration Workbench User Interface that displays the properties of a migration entity that has been selected in one of the tree views in the Navigator Pane.

Progress Window is the part of the Migration Workbench User Interface that contains informational, error, or warning messages describing the progress of the migration process.

Software Development Kit (SDK) is a set of well-defined application programming interfaces (APIs) that provide services that a software developer can use.

Source Database is the database containing the data dictionary of the application system being migrated by the Migration Workbench. The source database is a database other than Oracle, for example, MySQL.

Source Model is a replica of the data dictionary of the source database. It is stored in the Oracle Migration Workbench Repository and is loaded by the Migration Workbench with the contents of the data dictionary of the source database.

Workbench Repository is the area in an Oracle database used to store the persistent information necessary for the Migration Workbench to migrate an application system.

2

Migration Process

This chapter introduces the migration process by outlining the architecture of both MySQL and Oracle. It includes information on the following:

- MySQL Architecture
- Oracle Architecture
- Preparing for Migration
- Extending the Application
- Using Offline Data Loading

MySQL Architecture

MySQL is an open-source relational database management system (RDBMS). It uses a client/server architecture and is a multi-threaded, multi-user database server. It was designed specifically to be a fast server; therefore, it does not provide many of the features provided by other relational database systems, such as subselects, foreign keys, referential integrity, stored procedures, triggers, or views.

The access privilege system used by MySQL is designed to give comprehensive security to the data; however, it does require some configuration. The server provides concurrency control so different users cannot modify the same data at the same time. The locking mechanism is not adequate for tables with a lot of write actions from different users at the same time.

MySQL provides a number of client tools, the most commonly used are:

- `mysql` - An interactive client that allows you to issue queries on databases and view the results
- `mysqldump` - A tool that allows you to extract the schema and data within a MySQL database and place into a file

- `mysqlimport` - A tool that allows you to read the schema and data from a file and place into a MySQL database
- `mysqladmin` - A tool that allows you to perform administrative tasks, such as creating databases and dropping databases

MySQL is free to use for many different platforms, including Linux. It has ODBC support for Win32. It also has APIs for C, C++, Java, Perl, Python, and PHP.

Oracle Architecture

Oracle9*i*, Oracle8*i*, and Oracle8 databases are powerful, flexible, and scalable relational database management system (RDBMS) servers, that run on a range of computer systems, from personal computers to the largest mainframes. Oracle has been designed to run effectively in a client/server environment and supports hundreds to thousands of users.

The Oracle architecture supports advanced server features, such as record locking with versioning, advanced query optimization, the PL/SQL programming language, data replication, distributed database management, and other important features.

The architectural features described in this chapter are only a few of the features provided by Oracle. The features described are focused on the elements that pertain to working with MySQL. Refer to the following Oracle Server guides for a complete description of the Oracle architecture. These guides can also be found in online format on CD-ROM:

- *Oracle9i Database Concepts Release 1 (9.0.1)*
- *Oracle9i Database Administrator's Guide Release 1 (9.0.1)*
- *PL/SQL User's Guide and Reference Release 1 (9.0.1)*
- *Oracle9i Database Error Messages Release 1 (9.0.1)*

Triggers and Stored Procedures

Oracle allows you to write and store code in the DBMS along with data. You can associate trigger code with an UPDATE, INSERT, or DELETE event for each row or for a table as a whole. You can also set a trigger to run before or after the event. For example, you can set a trigger to run after any row is updated. This feature is not available in MySQL.

A stored procedure is a general routine, either function or subroutine, that is stored in pre-compiled form on the server. A trigger may call stored procedures, but triggers are only activated by specific database activity, such as the insertion of a row in a table.

PL/SQL Programming Language

The PL/SQL Programming Language is an ALGOL-based language like Pascal. PL/SQL is a modern, full-featured programming language with exception handling. You can use PL/SQL to write stored programs and triggers in Oracle. It is also the programming language used in many of the client-side tools available from Oracle, such as Forms from the Oracle Developer suite of products.

Sequences

A sequence is a unique number generator that is implemented in shared memory on a server. It is designed to provide a set of unique values for PL/SQL programs for use as primary keys. Sequences are designed for high performance applications that may otherwise ‘single-thread’ on table-based unique number generators.

Transactions

Oracle supports an implicit transaction model. Each SQL statement is part of a logical transaction. A logical transaction begins with the first SQL statement and ends with a COMMIT or ROLLBACK statement. Immediately after either of these statements, a new transaction takes effect with the next SQL statement.

Other Oracle Features

A database administrator has great flexibility when configuring Oracle. The administrator can write data on multiple disks for increased performance, tune rollback and recovery options, and allocate computer resources to optimize the configuration for each server. Oracle also supports distributed processing, so data can be distributed across multiple systems. Oracle offers a version of the server called Trusted Oracle Server for applications that require a higher level of user and use authentication.

Preparing for Migration

You must back up the MySQL database files before using the Migration Workbench Capture wizard to migrate to Oracle.

Extending the Application

After you move the data management portion of the MySQL application to Oracle, you can rely on Oracle to protect the data and maintain all referential integrity and business rules that you have encoded in PL/SQL.

With this foundation, you can extend the application with a wide range of tools. Oracle offers several high-productivity tools in the Oracle 9*i* Application Server, such as Oracle Portal or Oracle JDeveloper, and Oracle Objects for OLE.

In addition, if the application grows, you can move the Oracle server to larger computers without changing the application.

Using Offline Data Loading

You can use mysqldump, a client program shipped with the MySQL server, to output the schema and data of a MySQL database into .sql/.txt files in various formats. The Migration Workbench uses mysqldump in conjunction with SQL*Loader to provide an offline data loading capability for large tables. The following topics explain the process of offline data loading:

- Script Directory Structure
- Generating mysqldump Data Extract Scripts
- Using the Extracted Scripts

Script Directory Structure

The `%ORACLE_HOME%\Omwb\sqlloader_scripts` directory stores all data extraction scripts. Within this directory there is a sub-directory called MySQL that stores the SQL*Loader script output for MySQL. Within the MySQL directory, the Migration Workbench creates a directory using <timestamp> that represents the date and time you generated the SQL*Loader scripts. For example, a sub-directory called 1-10-99_17-56-16 indicates that you generated the scripts at 17:56 P.M. on 1st of October 1999.

As part of the Generate SQL*Loader Script command, a subfolder called Oracle is created in <timestamp> directory. The Oracle directory contains SQL*Loader control files and a SQL*Loader script called sql_load_script.bat. The SQL*Loader control files and the data files that you create should be located in this directory. Therefore, you should copy the dump_extract.bat file into the Oracle directory before executing the sql_load_script.bat file.

Generating mysqldump Data Extract Scripts

To create the mysqldump data extraction script and the SQL*Loader control files for all tables:

1. From the Oracle Model, click the **Tables** folder in the Oracle Model.
2. Choose **Object** -> **Generate SQL*Loader Scripts**.

> **Note:** You can also generate the scripts for a specific table by highlighting that table from the Oracle Model, then choosing **Object** -> **Generate SQL*Loader Scripts**.

3. If you are sure you want to generate the SQL*Loader scripts for the tables specified, click **Yes**.
4. After noting the location of the SQL*Loader scripts, click **OK**.

Using the Extracted Scripts

After generating the SQL*Loader scripts, you can use them to load the data into the Oracle database.

A description of the command line for each table in the dump_extract.bat file is as follows:

Code	Description
HOST	Host name of the system connecting to the MySQL server
USERNAME	MySQL user name
PASSWORD	Password of the MySQL user
`-T`	option for the mysql dump creates a .sql file that contains the SQL CREATE commands for the given table as well as a .txt file that contains the data for the given table.
`-fields-terminated -by=<ec>`	Used with the –T option to give a value to use as the closing value for each column - *ec* ends each column data for each row.

To use the scripts within the data extraction directories to execute a manual data extraction:

1. Add the host name, user name, and password to the dump_extract.bat file for connection to the MySQL server. This is indicated by the following tags: *`HOST, USERNAME, PASSWORD`*.

> **Note:** You must use a root user name and password or another user with DBA privileges.

2. Specify the output for the dump_extract.bat files by adding the destination path in the `<DESTINATION_PATH>` section. For instance, the destination path would be indicated as `%ORACLE_HOME%\Omwb\sqlloader_scripts\MySQL\<timestamp>\Oracle`.
3. Run the dump_extract.bat file to generate the data files. This extracts the data from the tables into data files called <table_name>.txt

> **Note:** Running the dump_extract.bat file also creates the <table_name>.sql files. However, they are not needed because the .CTL files also contain the table schema.

4. Move the data files into the destination path that you specified in the `<DESTINATION_PATH>` section of the dump_extract.bat file, such as `%ORACLE_HOME%\Omwb\sqlloader_scripts\MySQL\<timestamp>\Oracle` directory.
5. If the destination Oracle database does not reside on the same system as the Migration Workbench, you should FTP the entire destination directory to the same system as the destination Oracle database.
6. Execute the sql_load_script.bat file from the `%ORACLE_HOME%\Omwb\sqlloader_scripts\MySQL\<timestamp>\Oracle` directory to insert data files into the equivalent Oracle tables.

> **Note:** The migration of BLOBS is not supported by the scripted data move. You can remove the \n character is it exists in CLOBS.

3

MySQL Data Types, Reserved Words, and Operators

This chapter describes the data types used within Oracle. It shows the MySQL data types and what is the Oracle equivelent. It also provides you with a list of reserved words within Oracle. It includes information on the following:

- Supported Oracle Data Types
- Default Data Type Mappings
- Comparing MySQL to Oracle
- Oracle Reserved Words

Supported Oracle Data Types

Table 3–1 describes the Oracle data types supported by the Migration Workbench.

Table 3–1 Oracle Data Types Supported by Oracle Migration Workbench

Data Type	Description
BLOB	A binary large object. Maximum size is 4 gigabytes.
CHAR (SIZE)	Fixed-length character data of length size bytes. Maximum size is 2000 bytes. Default and minimum size is 1 byte.
CLOB	A character large object containing single-byte characters. Both fixed-width and variable-width character sets are supported, both using the CHAR database character set. Maximum size is 4 gigabytes.

Table 3–1 Oracle Data Types Supported by Oracle Migration Workbench

Data Type	Description
DATE	The DATE data type stores date and time information. Although date and time information can be represented in both CHAR and NUMBER data types, the DATE data type has special associated properties. For each DATE value, Oracle stores the following information: century, year, month, day, hour, minute, and second.
FLOAT	Specifies a floating-point number with decimal precision 38, or binary precision 126.
LONG (SIZE)	Character data of variable length up to 2 gigabytes, or 231 -1 bytes.
LONG RAW	Raw binary data of variable length up to 2 gigabytes.
NCHAR (SIZE)	Fixed-length character data of length size characters or bytes, depending on the choice of national character set. Maximum size is determined by the number of bytes required to store each character, with an upper limit of 2000 bytes. Default and minimum size is 1 character or 1 byte, depending on the character set.
NCLOB	A character large object containing multibyte characters. Both fixed-width and variable-width character sets are supported, both using the NCHAR database character set. Maximum size is 4 gigabytes. Stores national character set data.
NUMBER	Number having precision p and scale s. The precision p can range from 1 to 38. The scale s can range from -84 to 127.
NVARCHAR2 (SIZE)	Variable-length character string having maximum length size characters or bytes, depending on the choice of national character set. Maximum size is determined by the number of bytes required to store each character, with an upper limit of 4000 bytes. You must specify size for NVARCHAR2.
RAW (SIZE)	Raw binary data of length size bytes. Maximum size is 2000 bytes. You must specify size for a RAW value.
VARCHAR (SIZE)	The VARCHAR data type is currently synonymous with the VARCHAR2 data type. Oracle recommends that you use VARCHAR2 rather than VARCHAR. In the future, VARCHAR might be defined as a separate data type used for variable-length character strings compared with different comparison semantics. The maximum size is 4000 and the minimum of 1 is the default.

Refer to *Oracle9i SQL Reference, Release 1 (9.0.1)* for more information about Oracle data types.

Default Data Type Mappings

Table 3–2 shows the default settings used by the Migration Workbench to convert data types from MySQL to Oracle. The Migration Workbench allows you to change the default setting for certain data types by specifying an alternative type. You can do this in the Capture Wizard or in the Data Type Mappings page of the Options dialog box.

Refer to the Oracle Migration Workbench Online Help for more information about changing the default data type mappings.

Table 3–2 Default Data Type Mappings Used by Oracle Migration Workbench

MySQL **Data Type**	**Oracle Data Type**
TINYINT	NUMBER(3, 0)
SMALLINT	NUMBER(5, 0)
MEDIUMINT	NUMBER(7, 0)
INT	NUMBER(10, 0)
INTEGER	NUMBER(10, 0)
BIGINT	NUMBER(19, 0)
FLOAT	FLOAT
DOUBLE	FLOAT (24)
DOULBE PRECISION	FLOAT (24)
REAL	FLOAT (24)
DECIMAL	FLOAT (24)
NUMERIC	NUMBER
DATE	DATE
DATETIME	DATE
TIMESTAMP	NUMBER
TIME	DATE
YEAR	NUMBER
CHAR	CHAR
VARCHAR	VARCHAR2
TINYBLOB	RAW

Table 3–2 Default Data Type Mappings Used by Oracle Migration Workbench

MySQL **Data Type**	**Oracle Data Type**
TINYTEXT	VARCHAR2
BLOB	BLOB, RAW
TEXT	VARCHAR2, CLOB
MEDIUMBLOB	BLOB, RAW
MEDIUMTEXT	RAW, CLOB
LONGBLOB	BLOB, RAW
LONGTEXT	RAW, CLOB
ENUM	VARCHAR2, set to 100 by default
SET	VARCHAR2, set to 100 by default

Note: The Enum data type has no direct mapping in Oracle. The Migration Workbench maps Enum columns in MySQL to Varchar2 columns in Oracle. It then adds a constraint to those columns to ensure that only values that were allowed by the Enum data type are allowed in the column it was mapped to in Oracle.

The Set data type has no direct mapping in Oracle. The current version of the Migration Workbench maps Set columns in MySQL to Varchar2 columns in Oracle.

Comparing MySQL to Oracle

The following tables represent the mappings of the datatypes between MySQL and Oracle. For some MySQL datatypes there is more than one alternative Oracle datatype. The tables include information on the following:

- Numeric Types
- Date and Time Types
- String Types

Numeric Types

In the case of MySQL data types that map to numeric datatypes in Oracle the following conditions apply:

- If there is no precision or scale defined for the destination Oracle data type then precision and scale are taken from the MySQL source data type.
- If there is a precision or scale defined for the destination data type then these values are compared to the equivalent values of the source data type and the maximum value is selected.

The following table compares the numeric types of MySQL to Oracle:

MySQL	Size	Oracle
TINYINT	1 Byte	NUMBER(3,0)
SMALLINT	2 Bytes	NUMBER(5,0)
MEDIUMINT	3 Bytes	NUMBER (7,0)‘
INT	4 Bytes	NUMBER (10,0)
INTEGER	4 Bytes	NUMBER (10,0)
BIGINT	8 Bytes	NUMBER (19,0)
FLOAT(X<=24)	4 Bytes	FLOAT(0)
FLOAT(25<=X <=53)	8 Bytes	FLOAT(24)
DOUBLE	8 Bytes	FLOAT(24)
DOUBLE PRECION	8 Bytes	FLOAT(24)
REAL	8 Bytes	FLOAT(24)
DECIMAL	M Bytes(D+2, if M<D)	FLOAT(24)
NUMERIC	M Bytes(D+2, if M<D)	NUMBER

Date and Time Types

The following table compares the date and time types of MySQL to Oracle:

MySQL	Size	Oracle
DATE	3 Bytes	DATE
DATETIME	8 Bytes	DATE

MySQL	Size	Oracle
TIMESTAMP	4 Bytes	NUMBER
TIME	3 Bytes	DATE
YEAR	1 Byte	NUMBER

String Types

In the case of MySQL data types that map to character data types in Oracle, the following conditions apply:

- If there is no length defined for the destination data type then the length is taken from the source datatype.
- If there is a length defined for the destination data type then the maximum value of the two lengths is taken.

Note: Reference to M indicates the maximum display size. The maximum legal display size is 255. While a reference to L applies to a floating point types and indicates the number of digits following the decimal point.

The following compares the string types of MySQL to Oracle:

MySQL	Size	Oracle
CHAR(m)	M Bytes, 1<=M<=255	CHAR
VARCHAR(m)	L+1 Bytes whereas L<=M and 1<=M<=255	VARCHAR2
TINYBLOB	L + 1 Bytes whereas L<2 ^8	RAW, BLOB
BLOB	L + 2 Bytes whereas L<2^16	RAW, BLOB
TEXT	L + 2 Bytes whereas L<2^16	RAW, BLOB
MEDIUMBLOB	L + 3 Bytes whereas L < 2^ 24	RAW, BLOB
MEDIUMTEXT	L + 3 Bytes whereas L < 2^ 24	RAW, BLOB
LONGBLOB	L + 4 Bytes whereas L < 2 ^ 32	RAW, BLOB
LONGTEXT	L + 4 Bytes whereas L < 2 ^ 32	RAW, BLOB

MySQL	Size	Oracle
ENUM (VALUE1, VALUE2, ...)	1 or 2 Bytes depending on the number of enum. values (65535 values max)	
SET (VALUE1, VALUE2, ...)	1, 2, 3, 4 or 8 Bytes depending on the number of set members (64 members maximum)	

Oracle Reserved Words

The words are reserved in Oracle. The Migration Workbench appends an underscore to any object names that conflict with these reserved words.

ABORT

ACCEPT

ACCESS

ADD

ALL

ALTER

AND

ANY

ARRAY

ARRAYLEN

AS

ASC

ASSERT

ASSIGN

AT

AUDIT

AUTHORIZATION

AVG

BASE_TABLE

BEGIN

BETWEEN

BINARY_INTEGER

BODY

BOOLEAN

BY

CASE

CHAR

CHAR_BASE

CHECK

CLOSE

CLUSTER

CLUSTERS

COLAUTH

COLUMN

COMMENT

COMMIT

COMPRESS

CONNECT

CONSTANT

CRASH

CREATE
CURRENT
CURRVAL
CURSOR
DATA_BASE
DATABASE
DATE
DBA
DEBUGOFF
DEBUGON
DECIMAL
DECLARE
DEFAULT
DEFINITION
DELAY
DELETE
DESC
DIGITS
DISPOSE
DISTINCT
DO
DROP
ELSE
ELSIF
END
ENTRY
EXCEPTION
EXCEPTION_INIT
EXCLUSIVE
EXISTS
EXIT
FALSE
FETCH
FILE
FLOAT
FOR
FORM
FROM
FUNCTION
GENERIC
GOTO
GRANT
GROUP
HAVING
IDENTIFIED
IF
IMMEDIATE
IN
INCREMENT
INDEX
INDEXES
INDICATOR
INITIAL
INSERT
INTEGER
INTERFACE
INTERSECT
INTO
IS
LEVEL
LIKE
LIMITED

LOCK
LONG
LOOP
MAX
MAXEXTENTS
MIN
MINUS
MLSLABEL
MOD
MODE
MODIFY
NATURAL
NATURALN
NETWORK
NEW
NEXTVAL
NOAUDIT
NOCOMPRESS
NOT
NOWAIT
NULL
NUMBER
NUMBER_BASE
OF
OFFLINE
ON
ONLINE
OPEN
OPTION
OR
ORDER
OTHERS
OUT
PACKAGE
PARTITION
PCTFREE
PLS_INTEGER
POSITIVE
POSITIVEN
PRAGMA
PRIOR
PRIVATE
PRIVILEGES
PROCEDURE
PUBLIC
RAISE
RANGE
RAW
REAL
RECORD
REF
RELEASE
REMR
RENAME
RESOURCE
RETURN
REVERSE
REVOKE
ROLLBACK
ROW
ROWID
ROWLABEL

ROWNUM
ROWS
ROWTYPE
RUN
SAVEPOINT
SCHEMA
SELECT
SEPERATE
SESSION
SET
SHARE
SIGNTYPE
SIZE
SMALLINT
SPACE
SQL
SQLCODE
SQLERRM
START
STATEMENT
STDDEV
SUBTYPE
SUCCESSFUL
SUM
SYNONYM
SYSDATE
TABAUTH
TABLE
TABLES
TASK
TERMINATE
THEN
TO
TRIGGER
TRUE
TYPE
UID
UNION
UNIQUE
UPDATE
USE
USER
VALIDATE
VALUES
VARCHAR
VARCHAR2
VARIANCE
VIEW
VIEWS
WHEN
WHENEVER
WHERE
WHILE
WITH
WORK
WRITE
XOR

4 Troubleshooting

This chapter provides troubleshooting solutions and information on optimizing the command line options and avoiding issues connecting from the Migration Workbench to the MySQL server. It includes information on:

- Defining the User Account
- Dumping MySQL Data
- Optimizing Command Line Options

Note: For information on error messages, see the Error Messages topic in Appendix A of this guide.

Defining the User Account

When you installed the MySQL server onto the system, the root user, a user account with full DBA privileges, was set up by default. You must log on to the MySQL server through the Migration Workbench using this root user. This is because the Migration Workbench attempts to connect to the MySQL server as the root user. Therefore, in Step 1: Source Database Details of the Capture Wizard the Source Login ID field is set to root and is grayed out. Therefore, you can not modify this field to login as another user.

Dumping MySQL Data

If you are having difficulty migrating from MySQL to Oracle, you can contact the Migration Workbench support team. The Migration Workbench Development Team provides support and solutions to the migration problems. Choose **Help ->**

Online Support to view information that explains how to report a problem. This also provides a list of the support options available.

You can provide the support team with a dump of the MySQL database. This helps in tracking the problem and providing a swift solution. By using the `mysqldump` method to create a copy of the MySQL database you generate text files that are portable to other systems, even those with different hardware architecture. The Migration Workbench Development Team can regenerate the output into another database.

The table below provides an explanation of the code used to dump the MySQL data and to regenrate the database from the `mysqldump` output text file.

Command	Description
`mysqldump`	A tool that allows you to extract the schema and data in a MySQL database to a file.
`mysql`	Loads MySQL so you can carry out the command.
`-u` *`user name`*	The root MySQL user name. This user should have full DBA privileges.
`-p`*`password`*	The password of the root user of the MySQL database server
`--opt`	Optimizes table dumping speed and writes a dump file that is optimal for reloading speed. This option enables the `-add-drop-table`, `--add-locks`, `--all`, `--extended-insert`, `--quick` and `-lock-tables` options. For a list of definitions of the options enabled by `--opt`, refer to the Optimizing Options for MySQL section.
`database_name`	The name of the database containing the information you want to dump to an output text file.
`<`	Symbol used for re-directing the input in UNIX and NT.
`file_name.sql`	File name containing the MySQL database information.

To dump the MySQL data, use the following command:

```
% mysqldump -u user name -ppassword --opt database_name < file_name.sql
```

To regenerate the database from the `mysqldump` output text file into a database, use the following command:

```
% mysql -u user name -ppassword database_name < file_name.sql
```

Optimizing Command Line Options

You automatically switch on options within the `mysqldump` command line by using `--opt`. For more information on dumping the MySQL data, refer to the Dumping MySQL Data section. The commands encompassed by the -opt commands are:

Command	Description
`--add-drop-table`	Adds a `DROP TABLE IF EXISTS` statement before each `CREATE TABLE` statement.
`--all`	Includes all of the MySQL specific create options.
`--extended-insert`	Writes multiple row `INSERT` statements.
`--quick`	Dumps directly to the standard output without buffering the query. If you suspend `mysqldump` while using this option, you may interfere with other clients because it could cause the server to wait.
`--lock-tables`	Locks all tables as read only.

A

MySQL Error Messages

This appendix provides error messages that you may receive while migrating the MySQL database to Oracle.

Error Messages

As you use the Migration Workbench to migrate the data, the following error messages can occur:

Error Message	Solution
Cannot connect to MySQL server <*name*>. Is there a MySQL server running on the system / port you are trying to connect to?	You may see this error message during Step 1: Source Database Details of the Capture Wizard. This error can be caused by the following: ■ The source port number is set to 3306 by default. This port number refers to the port through which MySQL communicates. This is the default port that the MySQL installer uses. If the MySQL server was configured to use a different port, replace the 3306 value with the correct port value in the Source Port Number fields in Step 1: Source Database Details of the Capture wizard. ■ The Data Source Name field refers to the name of the system where the MySQL server resides. Make sure you enter the correct values in the Data Source Name.
Communication link failure: bad handshake	You may see this error message during Step 1: Source Database Details of the Capture wizard. The Source Password field refers to the MySQL password for the root user. Specify the source password correctly in Step 1: Source Database Details of the Capture wizard to avoid this error.

Error Message	Solution
Failed to create index <*name*>: java.sql.SQLException:ORA-01408: such column list already indexed.	You may see this error message during the migration phase. It occurs because you can declare a column or set of columns in MySQL as the primary key index and as a unique key index. In SQL*Plus, you can declare a column or set of columns as either primary or unique, not both. If the Migration Workbench has already migrated a set of columns declared as a primary key in a MySQL table, then discovers the same set of columns declared as a unique index, it is flagged as an error.

Index