易鲸捷

QianBase 数据库迁移指南 1.6.6

2020 年 12 月

版权

© Copyright 2019-2020 贵州易鲸捷信息技术有限公司

公告

本文档包含的信息如有更改，恕不另行通知。

保留所有权利。除非版权法允许，否则在未经易鲸捷预先书面许可的情况下，严禁改编或翻译本手册的内容。易鲸捷对于本文中所包含的技术或编辑错误、遗漏概不负责。

易鲸捷产品和服务附带的正式担保声明中规定的担保是该产品和服务享有的唯一担保。本文中的任何信息均不构成额外的保修条款。

声明

Microsoft® 和 Windows® 是美国微软公司的注册商标。Java® 和 MySQL® 是 Oracle 及其子公司的注册商标。Bosun 是 Stack Exchange 的商标。Apache®、Hadoop®、HBase®、Hive®、openTSDB®、Sqoop® 和 Trafodion® 是 Apache 软件基金会的商标。Esgyn，EsgynDB 和 QianBase 是易鲸捷的商标。

目录

前言

修订历史

版本	日期	描述
1.6.6	2020 年 12 月	版本升级
1.1.2	2019 年 9 月	新增钱库 Oracle 兼容性功能说明
1.1.0	2019 年 7 月	本文介绍如何将常见关系型数据库（Oracle、MySQL、SQL server 等）迁移至 QianBase。

本文简介

本指南介绍了不同数据库之间的数据迁移，包含从其他数据源迁移到 QianBase 中，以及从 QianBase 迁移到其他数据库目标。

目标读者

本指南的目标读者为 QianBase 系统管理员、开发人员和最终用户

相关文档

本指南为 QianBase 文档库的一部分，QianBase 文档库**包括但不限于**以下文档:

文档名称	说明

QianBase 安装部署指南	本文介绍安装 QianBase，包括安装前准备、安装 Hadoop 发行版、故障排除、配置、启用安全功能、提高安全性和卸载 QianBase 等。
易鲸捷Designer 用户指南	本文介绍易鲸捷图形化数据库管理工具
易鲸捷迁移工具用户指南	本文介绍如何安装和使用易鲸捷迁移工具。
QianBase 技术白皮书	本文介绍 QianBase 技术架构，组件介绍，技术特点等。
QianBase 数据库规划文档	本文介绍节点数量规划、数据目录和安装部署目录规划、集群角色分配规划等。
QianBase 管理员手册	本文介绍 QianBase 的日常运维常用系统命令、常用检查 SQL，用户权限配置，连接设置等内容。
QianBase 常见问题提排查与解决	本文介绍如何排查和解决 QianBase 的常见问题。
QianBase 灾难恢复手册	本文介绍 QianBase 灾难恢复设计原理，方案建议以及使用手册。
QianBase 备份恢复手册	本文介绍 QianBase 备份恢复设计原理，方案建议以及使用手册。
QianBase 数据库扩容指南	本文介绍 QianBase 如何更换节点，增加节点，删除节点等操作。
QianBase 数据库参数调优建议	本文介绍如何进行数据模型优化，CQD 参数优化等 。

QianBase 客户端安装手册	本文介绍 QianBase JDBC， ODBC 以及 Trafci 驱动安装。
QianBase JDBC 程序员参考指南	本文介绍 QianBase JDBC 驱动连接设置，开发人员指南。
QianBase ODBC 程序员参考指南	本文介绍 QianBase ODBC 驱动连接设置，开发人员指南。
QianBase SPSQL 存储过程用户手册	本文介绍 QianBase SPSQL 存储过程的使用。
Esgyn DBManager 用户手册	本文介绍图形化数据库监控运维工具 DB Manager 的使用。
QianBase 数据库迁移指南	本文介绍如何将常见关系型数据库（Oracle、MySQL、SQL server 等）迁移至 QianBase。
QianBase SQL 用户手册	本文是 QianBase 的 SQL 使用手册。

批评与建议

我们支持您对本指南做出的任何批评与建议，并尽力提供符合您需求的文档。

若您发现任何错误、或有任何改进建议，请发邮件至 support@esgyn.cn。

1. 数据库迁移概述

1.1. 迁移概述

不同数据库之间的数据迁移是一个复杂的系统工程，包括视图、索引、存储过程、函数、包、序列和触发器等数据库对象的元数据和数据迁移，元数据包含数据库对象的 DDL 定义，数据迁移包含从其它数据源迁移数据到 QianBase 中，以及从 QianBase 迁移数据到其它目标数据库。数据从源到目标的过程中往往还需要进行数据的清洗和必要的转换。

QianBase 支持 SQL92 ANSI/ISO 标准,支持主要的 SQL99 和 SQL2003 标准,可以很方便的进行元数据的迁移。

QianBase 支持从各种主流数据源进行数据迁移,也提供高效的数据迁移工具。QianBase 可以进行简单的数据清洗，更加复杂的数据清洗工作则需要利用 QianBase 的 JDBC/ODBC 接口开发专门的程序或使用专业的 ETL 软件，易鲸捷也提供统一的大数据治理平台,能够支持数据的迁移、清洗和数据的可视化展示。

1.2. QianBaseANSI 标准和扩展语句

QianBase 的 SQL 最符合 SQL 92 核心标准，此外 QianBase 的 SQL 还包括 SQL 99 标准的一些特性和部分 SQL 2003 标准，以及特殊的 QianBaseSQL 扩展语句。

1.2.1. 符合 ANSI 标准的语句

以下为符合 ANSI 标准、但某些包含 QianBase SQL 扩展语句 :

功能描述	SQL 语句
修改表	ALTER TABLE 语句
存储过程调用	CALL 语句
事务提交	COMMIT WORK 语句
创建函数	CREATE FUNCTION 语句
创建过程	CREATE PROCEDURE 语句
创建角色	CREATE ROLE 语句
创建模式	CREATE SCHEMA 语句
创建序列	CREATE SEQUENCE 语句
创建表	CREATE TABLE 语句
创建触发器	CREATE TRIGGER 语句
创建视图	CREATE VIEW 语句
删除数据	DELETE 语句
删除函数	DROP FUNCTION 语句
删除过程	DROP PROCEDURE 语句
删除角色	DROP ROLE 语句
删除模式	DROP SCHEMA 语句
删除序列	DROP SEQUENCE 语句
删除表	DROP TABLE 语句
删除触发器	DROP TRIGGER 语句
删除视图	DROP VIEW 语句
执行语句	EXECUTE 语句

授权语句	GRANT 语句
授权角色	GRANT ROLE 语句
插入语句	INSERT 语句
MERGE 语句	MERGE 语句
SQL 预编译	PREPARE 语句
删除权限	REVOKE 语句
删除角色	REVOKE ROLE 语句
回退事务	ROLLBACK WORK 语句
查询语句	SELECT 语句
模式设置	SET SCHEMA 语句
事务模式设置	SET TRANSACTION 语句
表语句	TABLE 语句
更新语句	UPDATE 语句
值语句	VALUES 语句

1.2.2. QianBase SQL 扩展语句

以下为符合 ANSI 标准的 QianBase SQL 扩展语句:

功能描述	SQL 语句
修改库	ALTER LIBRARY 语句
修改用户	ALTER USER 语句
开启事务模式	BEGIN WORK 语句
取消执行任务	CONTROL QUERY CANCEL 语句

查看 CQD 设置	CONTROL QUERY DEFAULT 语句
创建索引	CREATE INDEX 语句
创建库	CREATE LIBRARY 语句
删除索引	DROP INDEX 语句
删除存储库	DROP LIBRARY 语句
查看 SQL 执行计划	EXPLAIN 语句
GET 信息	GET 语句
查看 HBASE 对象	GET HBASE OBJECTS 语句
元数据版本	GET VERSION OF METADATA 语句
软件版本	GET VERSION OF SOFTWARE 语句
授权组件权限	GRANT COMPONENT PRIVILEGE 语句
生成一条记录描述	INVOKE 语句
数据装载	LOAD 语句
注册用户	REGISTER USER 语句
删除组件权限	REVOKE COMPONENT PRIVILEGE 语句
显示 CQD 属性	SHOWCONTROL 语句
查看表结构定义	SHOWDDL 语句
查看模式定义	SHOWDDL SCHEMA 语句
显示表中一组或多组列的直方图	SHOWSTATS 语句

统计信息	
卸载数据	UNLOAD 语句
删除用户注册	UNREGISTER USER 语句
更新统计信息	UPDATE STATISTICS 语句
批量插入数据	UPSERT 语句

1.2.3. 符合 ANSI 标准的函数

以下为符合 ANSI 标准、但某一些包含 QianBase SQL 扩展语句：

功能描述	SQL 语句
求平均	AVG 函数
条件判断	CASE（条件）表达式
数据类型转换	CAST 表达式
字符串长度	CHAR_LENGTH 函数
表达式判断	COALESCE 函数
统计函数	COUNT 函数
返回本地当前日期和时间的数值	CURRENT 函数
当前日期	CURRENT_DATE 函数
当前时间	CURRENT_TIME 函数
当前时间戳	CURRENT_TIMESTAMP 函数
当前用户	CURRENT_USER 函数

从表达式里抽取一个日期时间字段，返回一个精确数字型数值	EXTRACT 函数
返回小写字母数字字符	LOWER 函数
最大值函数	MAX/MAXIMUM 函数
最小值函数	MIN 函数
比较两个表达式的值	NULLIF 函数
返回以字节为单位的字符串长度	OCTET_LENGTH 函数
返回子串在字符串中的位置	POSITION 函数
当前会话用户	SESSION_USER
字符串操作	SUBSTRING/SUBSTR 函数
求和函数	SUM 函数
删除一个字符串的前导和尾随字符	TRIM 函数
返回大写字母数字字符	UPPER 函数

所有其它函数都是 QianBase SQL 扩展语句。

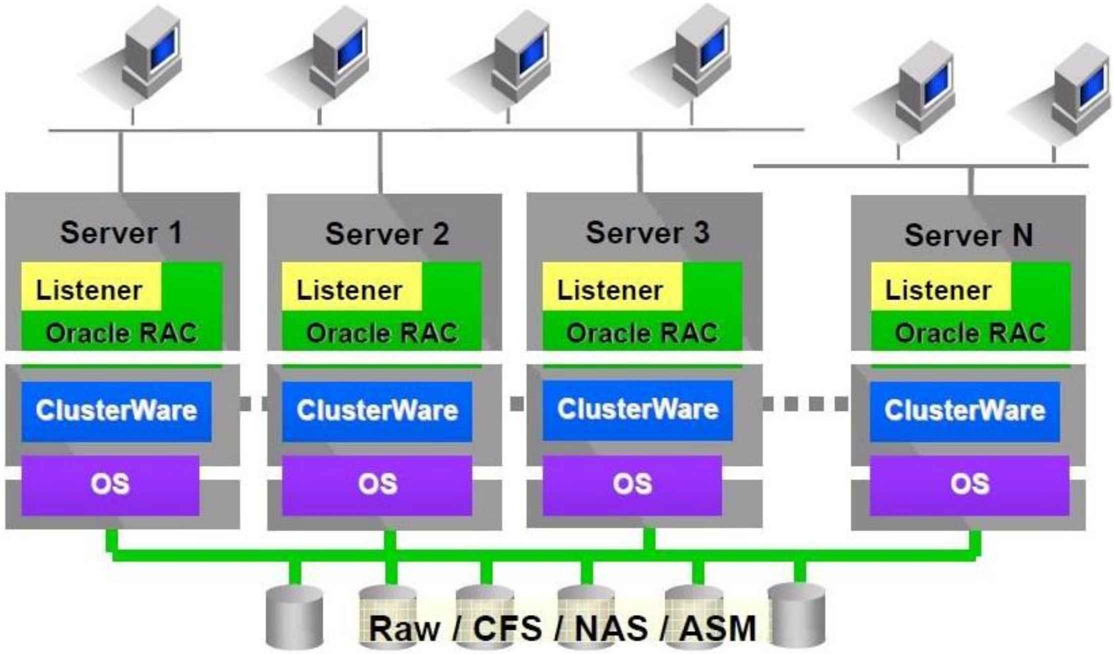
2. Oracle 数据库迁移

2.1 QianBase 和 Oracle 架构

ORACLE 数据库系统是美国 ORACLE 公司（甲骨文）提供的以 RAC 数据库

为核心的一组软件产品，是目前最流行的客户/服务器(CLIENT/SERVER)共享存储体系架构的集中式数据库之一。ORACLE 数据库是目前世界上使用最为广泛的集中式数据库管理系统，作为一个通用的数据库系统，它具有完整的数据管理功能;作为一个关系数据库，它是一个完备的关系数据库产品;作为集群数据库，它实现了 RAC 处理功能。

Oracle RAC（ Real Application Clusters ）的典型部署架构如下图所示：



QianBase 是企业级分布式关系数据库引擎，提供融合架构，针对大数据事务型或分析型的工作负载。事务型 SQL 包括 OLTP（联机事务处理）工作负载，支持传统企业级事务型应用程序（如 ERP、CRM 等）和企业业务流程产生的前面所述的工作负载。此外，业务工作负载可能包括业务报告以及托管和企业业务数据存储(ODS)的查询和数据分析任务。QianBase 是一个企业级的 SQL DBMS，能提供所有传统商业 RDBMS 为用户提供的服务。和传统数据库的区别在于，QianBase 基于 Hadoop/HBase 构建，能够提供极佳的水平扩展能力。当用户数据

量增加时，只需增加普通的计算机节点即可横向扩展存储和计算能力。

2.2 易鲸捷和 Oracle 数据库对象

Oracle 逻辑结构由至少一个表空间和数据库模式对象组成。这里，模式是对象的集合，而模式对象是直接引用数据库数据的逻辑结构。模式对象包括这样一些结构：表、视图、序列、过程、函数、包、同义词、索引、簇和数据库链等。逻辑存储结构包括表空间、段和范围，用于描述怎样使用数据库的物理空间。总之，逻辑结构由逻辑存储结构(表空间、段、范围、块)和逻辑数据结构(表、视图、序列、过程、函数、包、同义词、索引、簇和数据库链等)组成，而其中的模式对象(逻辑数据结构)和关系形成了数据库的关系设计。

QianBase 能够创建和管理传统关系型数据库对象，包括表、视图、序列、索引、二级索引、过程、函数、包和条件约束（check constraints），唯一约束 (unique constraints)，参照完整性约束（referential integrity constraints）。包括自我参照的外键（self-referential foreign keys），能确保上下级之间关系的一致性，比如分配给一个雇员的经理，确实作为另一个雇员存在。

	QianBase 对象	Oracle 对象
模式对象 Schema Object	Clusters	Clusters
	Constraints	Constraints
	UDF	Database links
	Database triggers	Database triggers
		Dimensions

	External procedure libraries	External procedure libraries
	Indexes	Index-organized tables
	Indexes	Indexes
		Indextypes
	Java classes, Java resources, Java sources	Java classes, Java resources, Java sources
		Materialized views
		Materialized view logs
		Mining models
	Libraries	
		Object tables
		Object types
		Object views
	Operators	Operators
	Packages	Packages
	Sequences	Sequences
	Functions	Functions
	Procedures	Procedures
		Synonyms
	Tables	Tables
	Views	Views
非模式对象 Nonschema Objects		Contexts
		Directories
		Editions
	Restore points	Restore points
	Roles	Roles

		Rollback segments
		Tablespaces
	Users	Users

2.3 数据类型

QianBase 支持的多种数据类型分配给列（表属性），包括数值，字符，字符串，日期，时间，时间间隔等等。 对用户数据和数据库元数据，通过 Unicode 编码如 UTF-8、UCS2 和 ISO 8859-1 实现国际标准（I18N）。 不同数据编码之间的比较和操作，由潜在的转换和翻译，透明地实现。

数据类型	QianBase 类型	Oracle 数据类型
字符串类型	CHARACTER(n) CHAR(n)	CHAR(n)
	VARCHAR CHARACTER VARYING(n) CHAR VARYING(n)	VARCHAR2(n)
	NATIONAL CHARACTER(n) NATIONAL CHAR(n) NCHAR(n)	NCHAR(n)
	NATIONAL CHARACTER VARYING(n) NATIONAL CHAR VARYING(n) NCHAR VARYING(n)	NVARCHAR2(n)
	VARCHAR	LONG 注：Oracle 的 LONG 类型最多可存储 2GB 数据，QianBase 的 VARCHAR 最多 16MB,超过 16MB

		使用 LOB 类型映射
数值型类型	NUMERIC[(p,s)] DECIMAL[(p,s)]	NUMBER(p,s)
	INT[EGER] SMALLINT LARGEINT NUMERIC(p,0) DECIMAL(p,0)	NUMBER(p,0) 注：Oracle 的 NUMBER 类型可能有小数位，也可能没有小数位，若无法确定是否有小数位，建议映射为 NUMERIC(p,s)
	FLOAT DOUBLE PRECISION REAL	FLOAT(126) FLOAT(126) FLOAT(63)
时间和日期类型	DATE TIMESTAMP	DATE(不带时分秒) DATE(带时分秒)
	TIME	
		BINARY_FLOAT
		BINARY_DOUBLE
	TIMESTAMP	TIMESTAMP
	INTERVAL	INTERVAL
大对象类型	CLOB	CLOB
	BLOB	BLOB
		NCLOB
		RAW
		LONG RAW

2.4 函数和功能

2.4.1 字符串函数

这些函数可以操作字符串，使用一个字符值表达式作为一个参数或返回一个

字符数据类型的结果。字符串函数将一个输入字符串里的单字节或多字节字符视为一个字符，不管该字符的字节长度如何。

QianBase	Oracle	函数说明
ASCII	ASCII	返回一个字符值表达式的首字符的 ASCII 代码值。
CHR/CHAR	CHR	返回一个字符集里的指定代码值。
CHAR_LENGTH		返回一个字符串里的字符数。您也可使用 CHARACTER_LENGTH 函数。
CODE_VALUE		返回一个无符号的整数，它是一个字符值表达式里的首字符的代码点，可以关联一种受支持的字符集。
CONCAT	CONCAT	返回两个字符值表达式的串联作为一个字符串值。您也可使用串联运算符（||）。
	INITCAP	将字符表达式的第一个字母大写，其余保持小写
INSTR	INSTR	字符串的子字符串查询功能
INSERT		返回一个字符串，其中指定数量的字符已被删除，并且另一个字符串被插入到指定的起始位置。
LCASE		降档字母数字字符。您也可使用 LOWER。

LEFT		返回一个字符表达式最左侧的指定数量的字符。
LENGTH	LENGTH	返回一个字符串里的字符数。您也可使用 CHAR_LENGTH 函数。
LOCATE		返回一个字符串里的指定子字符串所处的位置。您也可使用 POSITION 函数。与 Oracle 的 INSTR 功能类似。
LOWER	LOWER	降档字母数字字符。您也可使用 LCASE。
LPAD	LPAD	将一个字符表达式最左侧的指定数量的字符替换为一个填充字符。
LTRIM	LTRIM	删除一个字符串里的前导空格。
	NCHR	将字符编码转成字符
	NLS_INITCAP	将每个单词的首字母转成大写
	NLS_LOWER	转成小写
	NLS_UPPER	转成大写
OCTET_LENGTH	LENGTHB	返回一个字符串的长度，以字节为单位。
POSITION		返回字符串中指定子字符串所处的位置。您也可使用 LOCATE 函数。
REGEXP_REPLACE	REGEXP_REPLACE	通过正则表达式替换
REGEXP_SUBSTR	REGEXP_SUBSTR	通过正则表达式查找子字符串

REGEXP_COUNT	REGEXP_COUNT	通过正则表达式查找子字符串的数量
	REGEXP_INSTR	通过正则表达式查找子字符串的位置
REPEAT		返回一个字符串，其中包含按指定次数重复出现的字符表达式求值。
REPLACE	REPLACE	将原始字符串里出现的所有指定字符串都替换为另一个字符串后，再返回由此产生的字符串。
RIGHT		返回一个字符表达式最右侧指定数量的字符。
RPAD	RPAD	将一个字符表达式最右侧指定数量的字符替换为一个填充字符。
RTRIM	RTRIM	删除一个字符串里的尾随空格。
	SOUNDEX	返回发音一致的字符串
SPACE		返回一个含有指定数量非空格的字符串。
SUBSTRING/SUBSTR	SUBSTR	从一个字符串里抽取一个子字符串。
TRANSLATE	TRANSLATE	将一个字符串从源字符集转换为目标字符集。
TRIM	TRIM	删除一个字符串里的前导字符或尾随字符。
UCASE		升档字母数字字符。您也可使用 UPSHIFT 或 UPPER。

UPPER	UPPER	升档字母数字字符。您也可使用 UPSHIFT 或 UCASE。
UPSHIFT		升档字母数字字符。您也可使用 UPPER 或 UCASE。

2.4.2 日期时间函数

这些函数使用一个 Datetime 值表达式作为参数或返回一个数据类型为 Datetime 的结果:

QianBase	**Oracle**	
ADD_MONTHS	ADD_MONTHS	将 intr_expr 指定月份的整数值与 datetime_expr 相加，并将结果规范化。
	DBTIMEZONE	返回数据库时区
CONVERTTIMESTAMP		将一个 Julian 时间戳转换为 TIMESTAMP 值。
CURRENT		返回当前时间戳。您也可使用 CURRENT_TIMESTAMP 函数。
CURRENT_DATE	CURRENT_DATE	返回当前日期。
CURRENT_TIME		返回当前时间。
CURRENT_TIMESTAMP	CURRENT_TIMESTAMP	返回当前时间戳。您也可使用 CURRENT 函数。

DATE_ADD		将 interval 表达式指定的时间间隔与 datetime_expr 相加。
（一个 Interval）的 DATE_PART		从 Interval 指定的时间间隔值里抽取文本指定的 Datetime 字段，并返回结果作为精确的数字值。
（一个 Timestamp）的 DATE_PART		从 Timestamp 指定的日期时间值里抽取文本指定的 Datetime 字段，并返回结果作为精确的数字值。
DATE_SUB		从 datetime_expr 里减去指定的 interval_expression。
DATE_TRUNC		返回一个带有截断日时间部分的日期。
DATEADD		将 datepart 和 num_expr 指定的时间间隔与 datetime_expr 相加。
DATEDIFF		返回一个整数值作为起始日期和结束日期之间的 datepart 时间单元数量。
DATEFORMAT		格式化日期时间值（以便显

		示)。
DAY		返回一个整数值，代表该月的相应日，范围是 1 到 31。您也可使用 DAYOFMONTH 函数。
DAYNAME		从一个日期或时间戳表达式里返回一周中的一日的名称。
DAYOFMONTH		返回一个范围介于 1 和 31 之间的整数值，它代表一个月里的相应日。您也可使用 DAY。
DAYOFWEEK		返回一个整数值，它代表一周中的相应日，范围是 1 到 7。
DAYOFYEAR		返回一个整数值，它代表一年中的相应日，范围是 1 到 366。
EXTRACT	EXTRACT	从日期时间值表达式或时间间隔值表达式里返回指定的日期时间字段。
	FROM_TZ	将 TIMESTAMP 类型转

		成 TIMESTAMP WITH TIME ZONE 类型
HOUR		返回一个范围介于 0 和 23 之间的整数值，它代表一日中的相应小时。
JULIANTIMESTAMP		将一个日期时间值转换为儒历时间戳。
LAST_DAY	LAST_DAY	返回下一个指定的日期
	LOCALTIMESTAMP	返回当前 session 时区的日期时间
MINUTE		返回一个范围介于 0 和 59 之间的整数值，它代表一小时中的相应分钟。
MONTH		返回一个范围介于 1 和 12 之间的整数值，它代表一年中的相应月份。
MONTHS_BETWEEN	MONTHS_BETWEEN	返回这个日期之间的月份间隔
MONTHNAME		返回一个字符字面值，它代表一年中的月份名称（一月、二月等）。

NEXT_DAY	NEXT_DAY	返回下一个指定的日期从特定日期得到之后第一个星期几的日期。
	NEW_TIME	时区转换
QUARTER		返回一个范围介于 1 和 4 之间的整数值，它代表一年中的相应季度。
	ROUND	对 DATE 进行四舍五入
SYSDATE	SYSDATE	返回当前操作系统时区的日期
SYSTIMESTAMP	SYSTIMESTAMP	返回当前操作系统时区的日期时间
SECOND		返回一个范围介于 0 和 59 之间的整数值，它代表一分钟内的相应秒数。
TIMESTAMPADD		将 interval-ind（时间间隔指示）和 num_expr 指定的时间间隔加入 datetime_expr。
TIMESTAMPDIFF		返回一个整数值作为起始日期和结束日期之间的 interval-ind 单元数量。

TO_CHAR	TO_CHAR (datetime)	将大对象，日期，数字转成字符串
TO_TIMESTAMP	TO_TIMESTAMP	将字符串转成 TIMESTAMP
	TO_TIMESTAMP_TZ	将字符串转成 TIMESTAMP WITH TIME ZONE
	TO_YMINTERVAL	将字符串转成 INTERVAL YEAR TO MONTH
TRUNC	TRUNC (date)	对 DATE 进行四舍五入
	TZ_OFFSET	返回时差
WEEK		返回一个范围介于 1 和 54 之间的整数值，代表一年中相应的一周。
YEAR		返回一个代表年份的整数值。

2.4.3 数学函数

在一个 SQL 数字值表达式里使用以下数学函数:

QianBase 函数	**Oracle 函数**	**函数说明**

ABS	ABS	返回一个数字值表达式的绝对值。
ACOS	ACOS	返回一个数字值表达式的反余弦作为一个以弧度表示的角度。
ASIN	ASIN	返回一个数字值表达式的反正弦作为一个以弧度表示的角度。
ATAN	ATAN	返回一个数字值表达式的反正切作为一个以弧度表示的角度。
ATAN2	ATAN2	返回两个数字值表达式指定的 x 和 y 坐标的反正切作为一个以弧度表示的角度。
CEIL[ING]	CEIL	返回大于或等于一个数字值表达式的最小整数。
COS	COS	返回一个数字值表达式的余弦，这里的表达式是一个以弧度表示的角度。
COSH	COSH	返回一个数字值表达式的双曲余弦，这里的表达式是一个以弧度表示的角度。
DEGREES		将一个用弧度表示的数字值表达式转换为角度数。
EXP	EXP	返回一个数字值表达式的指数值（以 e 为底)。
FLOOR	FLOOR	返回小于或等于一个数字值表达式的最大整数。
	LN	返回参数的自然对数
LOG	LOG	返回一个数字值表达式的自然对数。
LOG10		返回一个数字值表达式的以 10 为底的对数。
MOD	MOD	返回一个整数值表达式除以另一个整数值表达式

		所得的余数（模数）。
	NANVL	如果参数 1 为 NaN，则返回参数 2
NULLIFZERO		返回操作数的值（如果它为 0，则将返回 NULL）。
PI		将 pi 常数值作为浮点值返回。
POWER	POWER	返回一个数字值表达式的（整数值表达式）次方得到的数值。您还可使用指数运算符**。
	REMAINDER	求余数
RADIANS		将一个用角度表示的数字值表达式转换为弧度数。
ROUND	ROUND	返回 numeric_expr 的数值，四舍五入到小数点右侧 num 位。
SIGN	SIGN	返回一个数字值表达式的符号标志。如果该数值小于 0，则将返回-1 作为标志。如果该数值等于 0，则将返回 0；如果该数值大于零，则将返回 1。
SIN	SIN	返回一个数字值表达式的正弦，这里的表达式是一个以弧度表示的角度。
SINH	SINH	返回一个数字值表达式的双曲正弦，这里的表达式是一个以弧度表示的角度。
SQRT	SQRT	返回一个数字值表达式的平方根。
TAN	TAN	返回一个数字值表达式的正切，这里的表达式是

		用弧度表示的角度。
TANH	TANH	返回一个数字值表达式的双曲正切，这里的表达式是一个以弧度表示的角度。
TRUNC	TRUNC	截取，不对数据做四舍五入。
ZEROIFNULL		返回操作数的值（它为 NULL，则将返回 0）。

2.4.4 转换函数

QianBase	**Oracle**	**说明**
	ASCIISTR	将字符串转成 ASCII 码
	BIN_TO_NUM	将二进制转成十进制
CAST	CAST	将一个值从一种类型转换为另一种数据类型。
	COMPOSE	将字符串转成 Unicode 码
	CONVERT	字符集转换
	DECOMPOSE	将 Unicode 码转成字符串
	HEXTORAW	将包含十六进制的字符串转成 RAW
	NUMTODSINTERVAL	将数字转成 INTERVAL DAY TO SECOND
	NUMTOYMINTERVAL	将数字转成 INTERVAL YEAR TO MONTH
	RAWTOHEX	将 RAW 转成十六进制的字符串

	RAWTONHEX	等价 于 TO_NCHAR(RAWTOHEX(raw))
	SCN_TO_TIMESTAMP	将 SCN 转成 TIMESTAMP
	TIMESTAMP_TO_SCN	将 TIMESTAMP 转成 SCN
	TO_BINARY_DOUBLE	将字符串或数字转 成 BINARY_DOUBLE
	TO_BINARY_FLOAT	将字符串或数字转 成 BINARY_FLOAT
	TO_BLOB	将 RAW 或 LONG RAW 转成 BLOB
TO_CHAR	TO_CHAR (character)	TO_CHAR with char operand
TO_CHAR	TO_CHAR (datetime)	TO_CHAR formats: DD-MON-YYYY, D, MM, YYYY, HH24, 9999.99
TO_CHAR	TO_CHAR (number)	TO_CHAR with numeric operand
	TO_CLOB	将字符串转成 CLOB
TO_DATE	TO_DATE	将字符串转成 DATE
	TO_DSINTERVAL	将字符串转 成 INTERVAL DAY TO SECOND
	TO_LOB	将 LONG 或 LONG RAW 转成 LOB
	TO_MULTI_BYTE	将单位字符转成多位字符
	TO_NCHAR (character)	将大对象，日期，数字转成本地字符 串
	TO_NCHAR (datetime)	将大对象，日期，数字转成本地字符 串

	TO_NCHAR (number)	将大对象，日期，数字转成本地字符串
	TO_NCLOB	将字符串转成 NCLOB
TO_NUMBER	TO_NUMBER	将字符串或数字转成 NUMBER
	TO_SINGLE_BYTE	将多位字符转成单位字符
TO_TIMESTAMP	TO_TIMESTAMP	将字符串转成 TIMESTAMP
	TO_TIMESTAMP_TZ	将字符串转成 TIMESTAMP WITH TIME ZONE
	TO_YMINTERVAL	将字符串转成 INTERVAL YEAR TO MONTH
	TREAT	将表达式转成指定类型
	UNISTR	将字符串转成 AL16UTF16 或 UTF8

2.4.5 加密和解密函数

QianBase	**Oracle 数据库**	**功能描述**
DECODE	DECODE	按照所提供的顺序将 expr 与每个 test_expr 的数值逐一进行比较。
	DUMP	返回参数的数据类型，长度，内部表示
	ORA_HASH	返回参数的 hash code
	VSIZE	返回参数内部表示长度

2.4.6 NULL 相关函数

QianBase	Oracle 数据库	功能描述
COALESCE	COALESCE	返回不含 NULL 值的列表里的第一个表达式的值；另外如果所有的表达式都具有 NULL 值，该函数也将返回一个 NULL 值。
	LNNVL	如果条件是假或未知，LNNVL 函数返回真
	NANVL	用来将 BINARY_FLOAT_NAN 和 BINARY_DOUBLE_NAN 转成其它值
NULLIF	NULLIF	如果两个操作数不相等，则返回第一个操作数的值；否则返回 NULL。
NVL	NVL	返回第一个操作数的值。如果它为 NULL，则将返回第二个操作数的值。
	NVL2	如果参数 1 为 NULL，则返回参数 3，否则返回 参数 2
ISNULL		返回第一个参数，如果它不为空；否则它将返回第二个参数。

2.4.7 聚合(集合)函数

一个聚合（集合）函数运行在通过查询语句检索到的一组或多组行上或存在聚合函数的子查询上。

QianBase	**Oracle 数据库**	**功能描述**
AVG	AVG	为函数表达式参数求值所派生的一组数字计算平均值。
	COLLECT	返回参数列的嵌套表
	CORR	计算皮尔逊相关系数
COUNT	COUNT	COUNT 是一种窗口函数，它为 inline-window specificatio 指定的当前窗口返回一个给定表达式的非空值的计数。
	COVAR_POP COVAR_SAMP	总体协方差 样本协方差
DENSE_RANK	DENSE_RANK	DENSE_RANK 是一种窗口函数，它返回 inline-window-specification 指定的当前分区里的每一行的排名。
	GROUP_ID GROUPING GROUPING_ID	用在 GROUPING SETS，ROLLUP□，CUBE 中
LAG	LAG	LAG 窗口函数返回一个分区中的当前行的前一行数值。如果不存在任何一行，则将返回 NULL。
	LAST	求最大值， 配合 DENSE_RANK 使用
LEAD	LEAD	LEAD 窗口函数返回一个分区中的当前行的后一行数值。如果不存在任何一行，则

		将返回 NULL。
LISTAGG	LISTAGG	连接列。您也可以使用 PIVOT 函数或 GROUP_CONCAT 函数。
MAX/MAXIMUM	MAX	从表达式参数求值所派生的一组数值中确定最大值。
	MEDIAN	连续分布模型
MIN	MIN	从表达式参数求值所派生的一组数值中确定最小值。
	PERCENT_RANK PERCENTILE_CONT PERCENTILE_DISC	和 CUME_DIST 类似 计算值的连续分布模型 计算值的不连续分布模型
RANK	RANK	RANK 是一种窗口函数，它返回 inline-window-specification 指定的当前分区里的每一行的排名。
ROW_NUMBER	ROW_NUMBER	ROW_NUMBER 是一种窗口函数，它返回 inline-window-specification 的当前窗口的每一行的行号。
STDDEV	STDDEV	STDDEV 从函数表达式参数求值所派生的一组数值中计算统计学标准差。这些数字可以加权处理。
STDDEV_POP	STDDEV_POP	STDDEV_POP 是一种窗口函数，它为

		inline-window-specification 指定的当前窗口返回给定表达式所有非空值的标准差。
SUM	SUM	从函数表达式参数求值所派生的一组数值中计算总和。
	VAR_POP	总体方差
	VAR_SAMP	样本方差
VARIANCE	VARIANCE	从函数表达式参数求值所派生的一组数值中计算统计方差。这些数字可以加权处理。

2.4.8 其它函数

QianBase	**Oracle 数据库**	**功能描述**
AUTHNAME		返回与特定授权 ID 号相关联的授权名称。
BITAND		在两个操作数的相应位上执行'AND'运算。
CASE (条件) 表达式		一个条件表达式。两种形式的 CASE 表达式分别为 CASE 简单表达式和 CASE 搜索表达式。
CONVERTTOHEX		将指定的值表达式转换为 16 进制数值，以便进行显示。
CURRENT_USER		返回调用该函数的当前用户的数据库用户

		名。
EXPLAIN		生成一个结果表，其描述 SELECT、INSERT、DELETE 或 UPDATE 语句的访问计划。
	UID	返回当前用户 ID
USER	USER	返回调用该函数的当前用户的数据库用户名或与指定的用户 ID 号相关联的数据库用户名。
	USERENV	

2.5 Oracle 迁移实践

QianBase 支持标准的 SQL 数据操作，支持 SELECT, INSERT, UPDATE, DELETE 和 UPSERT/MERGE 语法，并提供不同方式的 JOIN、UNION、WHERE、聚合（GROUP BY 和 HAVING）、排序、取样、相关和嵌套子查询、游标和很多 SQL 功能。

2.5.1 QianBase 兼容 Oracle 模式

建议设置 Oracle 兼容模式，便于更好的支持 Oracle 功能；

CQD MODE_COMPATIBLE_1 'ON';

2.5.2 Oracle 表达式数据类型转换

通常表达式不能包含不同数据类型的值。例如，表达式不能乘 5 乘以 10，然后追加'Jence'。但是，Oracle 支持从一个数据类型到另一个数据类型的隐式和显

式转换。

比如：SELECT salary + '10' FROM employees;

QianBase 同样可以实现的不同数据类型的隐式转换，SQL 不需要修改。

2.5.3 Oracle Sequence 对象

SEQUENCE 的语句和 Oracle 类似:

```
CREATE SEQUENCE sid/* INTERNAL */
    START WITH 1 /* NEXT AVAILABLE VALUE 1 */
    INCREMENT BY 1
    MAXVALUE 100000
    MINVALUE 1
    CACHE 25
    NO CYCLE
    LARGEINT;
```

下一个值函数: SEQNUM: SELECT SEQNUM(sid, NEXT) FROM DUAL;

当前值函数: SELECT SEQNUM (sid, CURRENT) FROM DUAL;

支持 Oracle 的 NEXTVAL 和 CURRVAL 语法，以获得当前和下一个序列数

2.5.4 Oracle DUAL 的语法支持

SELECT <expr> FROM DUAL;

2.5.5 Oracle PIVOT 功能

1) 这是一个群组聚合函数，用多列索引的方法把多行获取的列值转置到单行*

2) 与 MySQL GROUP_CONCAT 或者 Oracle LISTAGG 相同

3) 从 Oracle 的 XMLAGG/XMLELEMENT 语法，变为使用 PIVOT 语法

2.5.6 Oracle ROWNUM 和 ROWID 的语法支持

ROWNUM 和 ROWID 都是伪例 。

ROWNUM 是根据 SQL 查询出的结果给每行分配一个逻辑编号。ROWID 是物理结构上的,每条记录 INSERT 到数据库中,都会有一个唯一的物理记录。

QianBase 兼容 Oracle 的 ROWNUM 和 ROWID 语法。

select * from (select rownum rn, a.* from emp a) t where t.rn between 2 and 10;

2.5.7 Oracle 允许在 ORDER BY 里面使用表达式支持

允许在 ORDER BY 里面使用表达式，比如 ORDER BY a+1

2.5.8 Oracle Table/View 增强支持

支持‘CREATE OR REPLACE VIEW’功能

如果‘ALTER TABLE ADD PRIMARY KEY’在一张空表上完成，则优化为创建新表*

创建全局临时表 (与 VOLATILE 表相同)

2.5.9 Oracle UNIQUE 语法的支持支持

UNIQUE 语句在 SELECT 中 相当于 DISTINCT. (Ex: SELECT UNIQUE * FROM t)

2.5.10 Oracle 解析器支持

1) 从解析器去除所有保留关键字

2) 多个词现为非保留字（还没有覆盖所有在 Oracle 中的非保留关键字）

3) 在列定义中支持 ENABLE 和 NULL 语句

4) 自动转换 FLOAT 类型到 NUMERIC，DATE 到 TIMESTAMP(0)，VARCHAR2 到 VARCHAR

5) NUMERIC 精度默认为 18

6) BLOB/CLOB 类型映射 VARCHAR 时为 VARCHAR(16777216)

7) 用忽略的方式来处理 Oracle 提示(Ex: SELECT/*+ PARALLEL… */)

2.5.11 Oracle UNIQUE_ID 支持

QianBase 提供两种方式生成，SYS_GUID 及 UUID，如

SELECT sys_guid() FROM DUAL;

2.5.12 SELECT uuid() FROM DUAL;Oracle 时间日期函数

1) TO_CHAR，TO_DATE 不用格式化语句

2) TO_CHAR 支持 DATE、TIMESTAMP、CHAR 和 NUMERIC 类型

3) TO_CHAR 格式：DD-MON-YYYY, DD, MM, YYYY, HH24, MI, SS, 9999.99 等，详细格式请参考《QianBase SQL 参考手册（中文版).pdf》

4) DATE – DATE 结果单位为天，如 2019-01-01

5) TIMESTAMP – TIMESTAMP 结果单位为秒，如 2019-01-01 10:30:45，也可以精确到毫秒级别（最多 6 位），如 2019-01-01 10:30:45.123456

6) SYSDATE，SYSTIMESTAMP， TO_TIMESTAMP 函数支持

7) 支持不兼容的转换

2.5.13 Oracle 其它功能支持

使用“:<name>” 的参数处理为 “?<name>”, 使用:<number>的参数处理为无名参数

在双值上支持 GREATEST 和 LEAST

作为可选的子查询子句 (Ex: SELECT * FROM (SELECT * FROM t) WHERE…)

2.6. PL/SQL 存储过程迁移

PL/SQL 是 Oracle 数据库对 SQL 语句的扩展。在普通 SQL 语句的使用上增加了编程语言的特点，所以 PL/SQL 就是把数据操作和查询语句组织在 PL/SQL 代码的过程性单元中，通过逻辑判断、循环等操作实现复杂的功能或者计算的程序语言。

QianBase 支持 JAVA 存储过程（SPJ），用户可以在服务器上运行用 Java 编写的运营或分析型过程，供用户程序调用。 对于需要执行多个 SQL 语句，且每个语句都需要在用户和服务器端交换数据的过程，通过存储过程将处理放在服务器端，更加高效。

QianBase 也提供基于 HPL/SQL 实现存储过程功能。HPL/SQL 是一个开放源代码的工具（Apache 许可证 2）实现过程 SQL 语言 Apache Hive，SparkSQL，Impala 以及 Hadoop 实现其它任何 SQL，NoSQL 和 RDBMS。HPL/SQL 是一种混合异构语言理解语法和几乎任何现有的程序的 SQL 方言的语义，你可以使用任何数据库。HPL/SQL 语言是兼容 Oracle PL/SQL、 ANSI/ISO SQL / PSM（IBM DB2 , MySQL , Teradata 等）, Teradata BTEQ , PostgreSQL 的 PL/ pgsql（Netezza）, Transact-SQL（微软 SQL Server、Sybase），允许您利用现有的 SQL/DWH 技能和熟悉的方法在 Hadoop 实现数据仓库解决方案。这也有利于现有业务逻辑迁移到 Hadoop。

易鲸捷数据存储过程相关语句：

ALTER LIBRARY 语句	更新 QianBase 里的库对象的物理文件名。
CALL 语句	初始化易鲸捷 SQL 数据库里的存储过程。
CREATE FUNCTION 语句	将采用 C 语言编写的用户定义函数（UDF）注册为 QianBase 里的函数。
CREATE LIBRARY 语句	在 QianBase 里注册库对象。
CREATE PROCEDURE 语句	创建为 QianBase 里的存储过程。
DROP FUNCTION 语句	从 QianBase 里删除用户定义函数（UDF）。
DROP LIBRARY 语句	从 QianBase 里删除库对象。
DROP PROCEDURE 语句	从 QianBase 里删除存储过程。

存储过程示例：

```
/*易鲸捷 HPL/SQL 例程*/
CREATE OR REPLACE PROCEDURE proc1(IN   id_i INT,out v_msg STRING)
```

```
DECLARE
  num1 INT=10;

BEGIN
  PRINT '----------num1 is -----------';
  PRINT num1;
  SET v_msg_out = v_msg;
  EXEC 'CREATE TABLE IF NOT EXISTS test_sp (id INT,name CHAR(10))';
  EXEC 'INSERT INTO test_sp VALUES(1,"aa")';
  EXEC 'INSERT INTO test_sp VALUES(2,"bbb")';

 DECLARE cnt INT;
 EXECUTE 'SELECT COUNT(*) FROM test_sp' INTO cnt;
 PRINT '-------test_sp count is:-------';
 PRINT cnt;

 /*CURSOR Function*/
 FOR item IN  (SELECT id FROM test_sp WHERE  id <= id_i LIMIT 10  )
 LOOP
     PRINT('--------id is ---------');
     PRINT(item.id);
 END LOOP;

 EXEC 'DROP TABLE test_sp';

 PRINT '-------test_sp output:-------';
  PRINT v_msg_out;
```

END;

存储过程迁移

序号	差异项	差异及其存在问题	区别示例	
			QianBase	Oracle
1	执行参数	QianBase VARCHAR 类型参数需要指定长度	IN parameter VARCHAR（10）	Parameter IN VARCHAR
2	异常处理	定义	Exception1 condition；	Exception1 exception；
		Exception when others 和 Oracle 相同，其他方式需要修改为 declare handler 方式处理	Declare continue handler for exception1 Begin …… End (declare handler 需要放到对应语句块 begin 的下面)	Exception when exception then Begin …… End
		应用方式	Signal exception1; Resignal;	Raise exception1; Raise;
3	表左右关联	LEFT OUTER JOIN	SELECT * FROM T1 LEFT JOIN T2 ON T1.A=T2.A WHERE T1.B=T2.B	SELECT * FROM T1,T2 WHERE T1.A(+)=T2.A AND T1.B=T2.B
		RIGHT OUTER JOIN	SELECT * FROM T1 RIGTH　JOIN T2 ON T1.A=T2.A WHERE T1.B=T2.B	SELECT * FROM T1,T2 WHERE T1.A=T2.A(+) AND T1.B=T2.B

4	END LOOP	QianBase 不用跟后面的 label	END LOOP	END LOOP label1
5	动态 SQL	QianBase 对语句进行先 PREPARE 编译再 EXECUTE 执行	Sqlstr :=”select * from t1 where t1.a=? and t1.a=?”; Prepare s1 from sqlstr; Execute s1 using v1,v2;	Sqlstr :=”select * from t1 where t1.a=:1 and t1.b=:2”; Execute immediate sqlstr using v1,v2;
6	Open cursor for using	Open 游标方式不同	Declare cursor1(a int,b varchar) cursor is select * from t1 where t1.c1=a and t1.c2=b; Open cursor1(v1,v2)	Sqlstr :=”select * from t1 where t1.c1=:1 and t1.c2=:2”; Open s1 fro sqlstr using v1,v2;
7	动态游标	QianBase 在打开游标前不能有一个或多个 DML 语句，系统已自定义了动态游标类型 sys_refcursor	c_getpbcdtl sys_refcursor;	TYPE i_cursor_type IS REF CURSOR; c_getpbcdtl i_cursor_type;
8	用户自定义类型	QianBase 不支持自定义类型	不支持，无替代方案，需要修改相应函数和应用	CREATE OR REPLACE TYPE TY_STR_SPLIT IS TABLE OF VARCHAR2 (4000);

9	SQL%ROWCOUNT		ACTIVITY_COUNT	SQL%ROWCOUNT
10	SQLCODE	Oracle 定义 sqlcode 变量	QianBase 的内置变量，可以直接使用，不用定义 sqlcode 变量	Sqlcode 变量需要定义
11	函数，存储过程创建	QianBase 需要在创建语句前加上 set sqlterminator %开始 set sqlterminator；结束	Set sqlterminator % Create or prelace procedure p() is begin End; % Set sqlterminator;	Create or prelace procedure p() is begin End;
12	执行存储过程	QianBase 是 call 后跟存储过程名称	CALL procedure_name(prepare);	命令窗口 exec procedure_name(prepare); PL/SQL Pl/SQL 窗口: Begin Procedure_name(prepare); End
13	获取查询数据前几行	QianBase 用 limit	Select * from t1 limit 1;	Select * from t1 fetch first 1 row only;

14	避免关键字及其不能识别的字符		REAL 等是 QianBase 关键字，不能作为变量名或者表名，删除调 sql 中的中文全角空格字符。	

3. MySQL 数据库迁移

3.1. QianBase 和 MySQL 架构对比

MySQL 是一个关系型数据库管理系统，由瑞典 MySQLAB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

基础架构

单点（Single），适合小规模应用

复制（Replication），适合中小规模应用

集群（Cluster），适合大规模应用

MySQL 在大数据量的情况下，需要分库分表，保证数据库的访问性能，应用层面实现业务逻辑和不同数据源的访问;

QianBase，企业级 SQL 关系数据库引擎，提供融合架构，针对大数据事务性或分析的工作负载。事务性 SQL 包括 OLTP（联机事务处理）工作负载，支持传统企业级事务性应用程序（如 ERP、CRM 等）和企业业务流程产生的前面所述的工作负载。此外，业务工作负载可能包括业务报告以及托管和企业业务数据存储（ODS）的查询和数据分析任务。QianBase 是一个企业级的 SQL DBMS，能提供所有传统商业 RDBMS 为用户提供的服务。和传统数据库的区别在于，QianBase 基于 Hadoop/HBase 构建，能够提供极佳的水平扩展能力。当用户数据量增加时，只需增加普通的计算机节点即可横向扩展存储和计算能力。

3.2. 易鲸捷和 MySQL 数据库对象

MySQL 逻辑结构由至少一个表空间和数据库模式对象组成。这里，模式是对象的集合，而模式对象是直接引用数据库数据的逻辑结构。模式对象包括这样一些结构：表、视图、索引等。逻辑存储结构包括表空间，用于描述怎样使用数据库的物理空间。总之,逻辑结构由逻辑存储结构(表空间和逻辑数据结构(表、视图、索引等)组成,而其中的模式对象(逻辑数据结构)和关系 形成了数据库的关系设计。

QianBase 能够创建和管理传统关系型数据库对象，包括表、视图、序列、索引、二级索引、过程、函数、包和条件约束（check constraints），唯一约束 (unique constraints)，参照完整性约束（referential integrity constraints）。包括自我参照的外键（self-referential foreign keys），能确保上下级之间关系的一致性，比如分配给

一个雇员的经理，确实作为另一个雇员存在。

	QianBase 对象	MySQL 对象
模式对象 Schema Object		Database
	Table	Table
	Column	Column
	Index	Index
	Constraint	Constraint
	Stored Program	Stored Program
	View	View
	--	Tablespace
		Server
	--	Log File Group
		Alias
		Compound Statement Label
		User-Defined Variable

3.3. 数据类型

QianBase 支持的多种数据类型分配给列(表属性),包括数,字符,VARCHAR,日期，时间，时间间隔等等。对用户数据和数据库元数据，通过 Unicode 编码如 UTF-8，UCS2 和 ISO 8859-1 实现国际标准（I18N）。不同数据编码之间的比较和操作，由潜在的转换和翻译，透明地实现。

数据类型	QianBase 类型	MySQL 数据类型
数值型类型	NUMERIC[(p,s)] DECIMAL[(p,s)]	NUMERIC DECIMAL

3.MySQL 数据库迁移

	INTEGER INT TINYINT SMALLINT LARGEINT BIGINT	INT TINYINT SMALLINT MEDIUMINT BIGINT
	FLOAT DOUBLE PRECISION REAL	FLOAT DOUBLE PRECISION REAL
时间和日期类型	DATE	DATE
	TIME	TIME
		DATETIME
	TIMESTAMP	TIMESTAMP
		YEAR
	INTERVAL	
字符串类型	CHARACTER(n) CHAR(n)	CHAR
	VARCHAR CHARACTER VARYING(n) CHAR VARYING(n)	VARCHAR
	NATIONAL CHARACTER(n) NATIONAL CHAR(n) NCHAR(n)	
	NATIONAL CHARACTER VARYING(n) NATIONAL CHAR VARYING(n) NCHAR VARYING(n)	
	BLOB	BLOB
	CLOB	TEXT

	BINARY	BINARY
	VARBINARY	VARBINARY
		ENUM
		SET

3.4. 函数和功能

下表是 QianBase 和 MySQL 功能或者函数的对应关系。

3.4.1. 字符串函数

这些函数可以操作字符串，使用一个字符值表达式作为一个参数或返回一个字符数据类型的结果。字符串函数将一个输入字符串里的单字节或多字节字符视为一个字符，不管该字符的字节长度如何。

QianBase	MySQL	函数说明
ASCII	ASCII	返回一个字符值表达式的首字符的 ASCII 代码值。
CONVERTTOBITS()	BIN	返回二进制
	BINARY	将字符串转换为二进制字符串
BIT_LENGTH()	BIT_LENGTH()	返回字符串的比特长度
CHAR	CHAR	返回一个字符集里的指定代码值。
CHAR_LENGTH	CHAR_LENGTH CHARACTER_LENGTH	返回一个字符串里的字符数。您也可使用 CHARACTER_LENGTH 函数。

CODE_VALUE		返回一个无符号的整数,它是一个字符值表达式里的首字符的代码点，可以关联一种受支持的字符集。
CONCAT	CONCAT	返回两个字符值表达式的串联作为一个字符串值。您也可使用串联运算符（‖）。
	CONCAT_WS	将 s1,s2...,sn 连接成字符串，并用 sep 字符间隔
	ELT	返回索引字符串
	EXPORT_SET	返回一个字符串，每一位的值位，你会得到一个字符串，每个设置点，你会得到一个字符串
	FIELD	在参数中返回第一个参数的索引
	FIND_IN_SET	分析逗号分隔的 list 列表，如果发现 str，返回 str 在 list 中的位置
	FORMAT	返回格式化到指定小数位数的数字
	FROM_BASE64	解码 Base-64 字符串并返回结果
CONVERTTOHEX	HEX	返回一个十进制或十六进制表示的字符串值
INSERT	INSERT	返回一个字符串,其中指定数量的

		字符已被删除,并且另一个字符串被插入到指定的起始位置。
LOCATE	INSTR	返回字符串的第一个匹配项的索引
LCASE	LCASE	降档字母数字字符。您也可使用 LOWER。
LEFT	LEFT	返回一个字符表达式最左侧的指定数量的字符。
	LENGTH	返回字符串 str 中的字符数
LOCATE	LOCATE	返回一个字符串里的指定子字符串所处的位置。您也可使用 POSITION 函数。
LOWER	LOWER	降档字母数字字符。您也可使用 LCASE。
LPAD	LPAD	将一个字符表达式最左侧的指定数量的字符替换为一个填充字符。
LTRIM	LTRIM	删除一个字符串里的前导空格。
	MAKE_SET	返回一组逗号分隔的字符串
	MATCH	执行全文搜索
	MID	从指定位置回一个字符串
	OCT	返回一个字符串包含一个数字的八进制表示

OCTET_LENGTH	OCTET_LENGTH	返回一个字符串的长度,以字节为单位。
	ORD	最左边的字符返回字符编码
POSITION	POSITION	返回字符串中指定子字符串所处的位置。您也可使用 LOCATE 函数。
	QUOTE()	用反斜杠转义 str 中的单引号
REPEAT	REPEAT	返回一个字符串,其中包含按指定次数重复出现的字符表达式求值。
REPLACE	REPLACE	将原始字符串里出现的所有指定字符串都替换为另一个字符串后,再返回由此产生的字符串。
	REVERSE()	返回颠倒字符串 str 的结果
RIGHT	RIGHT	返回一个字符表达式最右侧指定数量的字符。
RPAD	RPAD	将一个字符表达式最右侧指定数量的字符替换为一个填充字符。
RTRIM	RTRIM	删除一个字符串里的尾随空格。
	SOUNDEX()	返回发音一致的字符串
SPACE	SPACE	返回一个含有指定数量非空格的字符串。
	STRCMP()	比较字符串 s1 和 s2

SUBSTRING/SUBSTR	SUBSTRING/SUBSTR	从一个字符串里抽取一个子字符串。
	SUBSTRING_INDEX()	从指定位置返回字符串的子串
	TO_BASE64()	返回转换成 Base-64 字符串
TRANSLATE		将一个字符串从源字符集转换为目标字符集。
TRIM	TRIM	删除一个字符串里的前导字符或尾随字符。
UCASE	UCASE	升档字母数字字符。您也可使用 UPSHIFT 或 UPPER。
UNHEX	UNHEX()	返回一个包含十六进制表示的字符串
UPPER	UPPER	升档字母数字字符。您也可使用 UPSHIFT 或 UCASE。
UPSHIFT		升档字母数字字符。您也可使用 UPPER 或 UCASE。
	WEIGHT_STRING()	返回字符串的权重字符串

3.4.2. 日期时间函数

这些函数使用一个 Datetime 值表达式作为参数或返回一个数据类型为 Datetime 的结果:

QianBase	MySQL	函数说明
DATE_ADD	ADDDATE	将时间值（间隔）添加到日期值
	ADDTIME	添加时间
ADD_MONTHS		将intr_expr指定月份的整数值与 datetime_expr 相加，并将结果规范化。
	CONVERT_TZ	返回数据库时区
CONVERTTIMESTAMP		将一个儒略历时间戳转换为 TIMESTAMP 值。
CURRENT		返回当前时间戳。您也可使用 CURRENT_TIMESTAMP 函数。
CURRENT_DATE	CURDATE CURRENT_DATE	返回当前日期。
CURRENT_TIME	CURRENT_TIME CURTIME	返回当前时间。
CURRENT_TIMESTAMP	CURRENT_TIMESTAMP	返回当前时间戳。您也可使用 CURRENT 函数。
DATE_PART	DATE	提取日期或日期时间表达式的日期部分
DATE_ADD	DATE_ADD	将 interval_表达式指定的时间

		间隔与 datetime_expr 相加。
（一个 Interval）的 DATE_PART		从 Interval 指定的时间间隔值里抽取文本指定的 Datetime 字段，并返回结果作为精确的数字值。
（一个 Timestamp）的 DATE_PART		从 Timestamp 指定的日期时间值里抽取文本指定的 Datetime 字段，并返回结果作为精确的数字值。
DATE_SUB	DATE_SUB	从 datetime_expr 里减去指定的 interval_expression。
DATE_TRUNC		返回一个带有截断日时间部分的日期。
DATEADD	ADDDATE	将 datepart 和 num_expr 指定的时间间隔与 datetime_expr 相加。
DATEDIFF	DATEDIFF	返回一个整数值作为起始日期和结束日期之间的 datepart 时间单元数量。
DATEFORMAT	DATE_FORMAT	格式化日期时间值（以便显示）。
DAY	DAY	返回一个整数值，代表该月的

		相应日，范围是 1 到 31。您也可使用 DAYOFMONTH 函数。
DAYNAME	DAYNAME	从一个日期或时间戳表达式里返回一周中的一日的名称。
DAYOFMONTH	DAYOFMONTH	返回一个范围介于 1 和 31 之间的整数值，它代表一个月里的相应日。您也可使用 DAY。
DAYOFWEEK	DAYOFWEEK	返回一个整数值，它代表一周中的相应日，范围是 1 到 7。
DAYOFYEAR	DAYOFYEAR	返回一个整数值，它代表一年中的相应日，范围是 1 到 366。
EXTRACT	EXTRACT	从日期时间值表达式或时间间隔值表达式里返回指定的日期时间字段。
	FROM_DAYS	从天转化为日期
	FROM_UNIXTIME	根据指定的 fmt 格式，格式化 UNIX 时间戳 ts
	GET_FORMAT	返回日期格式字符串
HOUR	HOUR	返回一个范围介于 0 和 23 之间的整数值，它代表一日中的相应小时。

JULIANTIMESTAMP		将一个日期时间值转换为儒历时间戳。
LAST_DAY	LAST_DAY	返回下一个指定的日期
	LOCALTIME LOCALTIME	now()的同义词
	LOCALTIMESTAMP	返回当前 session 时区的日期时间
	MAKEDATE	创建日期格式
	MAKETIME	创建时间
	MICROSECOND	返回微妙值
MINUTE	MINUTE	返回一个范围介于 0 和 59 之间的整数值，它代表一小时中的相应分钟。
MONTH	MONTH	返回一个范围介于 1 和 12 之间的整数值，它代表一年中的相应月份。
MONTHNAME	MONTHNAME	返回一个字符字面值，它代表一年中的月份名称（一月、二月等)。
NOW	NOW	返回当前的日期和时间
	PERIOD_ADD	返回前一个指定的日期
	PERIOD_DIFF	时区转换

QUARTER	QUARTER	返回一个范围介于 1 和 4 之间的整数值，它代表一年中的相应季度。
	SEC_TO_TIME	对 DATE 进行四舍五入
SYSDATE	SYSDATE	返回当前操作系统时区的日期
SYSTIMESTAMP		返回当前操作系统时区的日期时间
SECOND	SECOND	返回一个范围介于 0 和 59 之间的整数值，它代表一分钟内的相应秒数。
TO_DATE	STR_TO_DATE	将字符串转换为日期
SUBDATE	SUBDATE	DATE_SUB()的同义词
	SUBTIME	时间减倍
	TIME	提取表达式传递的时间部分
TO_TIME	TIME_FORMAT	时间转换
	TIME_TO_SEC	返回转换为秒
TIMEDIFF	TIMEDIFF	时间相减
TIMESTAMP	TIMESTAMP	返回时间戳格式
TIMESTAMPADD	TIMESTAMPADD	将 interval-ind（时间间隔指示）和 num_expr 指定的时间间隔

		加入 datetime_expr。
TIMESTAMPDIFF	TIMESTAMPDIFF	返回一个整数值作为起始日期和结束日期之间的 interval-ind 单元数量。
TO_CHAR		将大对象，日期，数字转成字符串
	TO_DAYS	将日期参数转换为天数
	TO_SECONDS	将日期或日期时间参数转换为 0 年以来的秒数
	UNIX_TIMESTAMP	返回 UNIX 时间戳
	UTC_DATE	返回当前 UTC 日期
	UTC_TIME	返回当前 UTC 时间
	UTC_TIMESTAMP	返回当前 UTC 时间戳
WEEK	WEEK	返回一个范围介于 1 和 54 之间的整数值，代表一年中相应的一周。
WEEKDAY	WEEKDAY	返回星期几（0 到 6）
WEEKOFYEAR	WEEKOFYEAR	返回第几周（52 周）
YEAR	YEAR	返回一个代表年份的整数值。
YEARWEEK	YEARWEEK	返回制定日期的年和周

3.4.3. 数学函数

在一个 SQL 数字值表达式里使用以下数学函数:

QianBase	**MySQL**	**函数说明**
ABS	ABS	返回一个数字值表达式的绝对值。
ACOS	ACOS	返回一个数字值表达式的反余弦作为一个以弧度表示的角度。
ASIN	ASIN	返回一个数字值表达式的反正弦作为一个以弧度表示的角度。
ATAN	ATAN	返回一个数字值表达式的反正切作为一个以弧度表示的角度。
ATAN2	ATAN2	返回两个数字值表达式指定的 x 和 y 坐标的反正切作为一个以弧度表示的角度。
CEILING	CEIL/CEILING	返回大于或等于一个数字值表达式的最小整数。
	CONV	在数之间转换数字
COS 函数	COS	返回一个数字值表达式的余弦，这里的表达式是一个以弧度表示的角度。
COT	COT	返回的余切
CRC	CRC32	计算循环冗余校验值
COSH		返回一个数字值表达式的双曲余弦，这里的表达式是一个以弧度表示的角度。
DEGREES	DEGREES	将一个用弧度表示的数字值表达式转换为角度数。

EXP	EXP	返回一个数字值表达式的指数值(以 e 为底)。
FLOOR	FLOOR	返回小于或等于一个数字值表达式的最大整数。
	LN	返回参数的自然对数
LOG	LOG	返回一个数字值表达式的自然对数。
LOG10	LOG10	返回一个数字值表达式的以 10 为底的对数。
LOG2	LOG2	返回基-2 对数
MOD	MOD	返回一个整数值表达式除以另一个整数值表达式所得的余数（模数)。
NULLIFZERO		返回操作数的值（如果它为 0，则将返回 NULL)。
PI	PI	将 pi 常数值作为浮点值返回。
POWER	POWER/POW	返回一个数字值表达式的(整数值表达式) 次方得到的数值。您还可使用指数运算符**。
RADIANS	RADIANS	将一个用角度表示的数字值表达式转换为弧度数。
RAND	RAND	返回一个随机浮点值
ROUND	ROUND	返回 numeric_expr 的数值，四舍五入到小数点右侧 num 位。
SIGN	SIGN	返回一个数字值表达式的符号标志。如果该数值小于 0，则将返回-1 作为标志。如果该数值等于 0，则将返回 0；如果该数值大于零，

		则将返回 1。
SIN	SIN	返回一个数字值表达式的正弦，这里的表达式是一个以弧度表示的角度。
SINH		返回一个数字值表达式的双曲正弦，这里的表达式是一个以弧度表示的角度。
SQRT	SQRT	返回一个数字值表达式的平方根。
TAN	TAN	返回一个数字值表达式的正切，这里的表达式是用弧度表示的角度。
TANH		返回一个数字值表达式的双曲正切，这里的表达式是一个以弧度表示的角度。
	TRUNCATE	四舍五入
ZEROIFNULL		返回操作数的值（它为 NULL，则将返回 0）。

3.4.4. 转换函数

QianBase	**MySQL**	**函数说明**
CAST	CAST	将一个值从一种类型转换为另一种数据类型。
TRANSLATE	CONVERT	将值作为某种类型
TO_CHAR		TO_CHAR with char operand
TO_CHAR		TO_CHAR formats: DD-MON-YYYY, D, MM, YYYY, HH24, 9999.99
TO_CHAR		TO_CHAR with numeric operand
TO_DATE		将字符串转成 DATE

3.4.5. 加密和解密函数

QianBase	MySQL	函数说明
DECODE	DECODE	按照所提供的顺序将 expr 与每个 test_expr 的数值逐一进行比较。
AES_DECRYPT	AES_DECRYPT	用 AES 加密
AES_ENCRYPT	AES_ENCRYPT	用 AES 解密
	COMPRESS	返回结果作为二进制字符串
	ENCODE	加密字符串
MD5	MD5	计算 MD5 验证码
RANDOM_BYTES	RANDOM_BYTES	返回随机字节向量
SHA1/SHA2	SHA1/SHA2	计算 SHA-2 验证码
	UNCOMPRESS	解压缩字符串
	UNCOMPRESSED_LENGTH	返回压缩前的长度
	VALIDATE_PASSWORD_STRENGTH	确定密码强度

3.4.6. NULL 相关函数

QianBase	MySQL	函数说明
COALESCE	COALESCE	返回不含 NULL 值的列表里的第一个表达式的值；另外如果所有的表达式都具有 NULL 值，该函数也将返回一

		个 NULL 值。
NULLIF	NULLIF	如果两个操作数不相等，则返回第一个操作数的值；否则返回 NULL。
NVL		返回第一个操作数的值。如果它为 NULL，则将返回第二个操作数的值。
ISNULL	ISNULL	返回第一个参数，如果它不为空；否则它将返回第二个参数。

3.4.7. 聚合(集合)函数

一个聚合（集合）函数运行在通过 SELECT 语句检索到的一组或多组行上或存在聚合函数的子查询上。

QianBase	**MySQL**	**函数说明**
AVG	AVG	为函数表达式参数求值所派生的一组数字计算平均值。
BIT_AND	BIT_AND	返回按位与
BIT_OR	BIT_OR	返回按位或
BIT_XOR	BIT_XOR	返回按位异或
COUNT	COUNT	COUNT 是一种窗口函数，它为 inline-window specificatio 指定的当前窗口返回一个给定表达式的非空值的计数。
DENSE_RANK		DENSE_RANK 是一种窗口函数，它返回 inline-window-specification 指定的当前分

		区里的每一行的排名。
GROUP_CONCAT	GROUP_CONCAT	用于把多列转成一行
LAG		LAG 窗口函数返回一个分区中的当前行的前一行数值。如果不存在任何一行，则将返回 NULL。
LEAD		LEAD 窗口函数返回一个分区中的当前行的后一行数值。如果不存在任何一行，则将返回 NULL。
MAX/MAXIMUM	MAX	从表达式参数求值所派生的一组数值中确定最大值。
MIN	MIN	从表达式参数求值所派生的一组数值中确定最小值。
RANK		RANK 是一种窗口函数，它返回 inline-window-specification 指定的当前分区里的每一行的排名。
ROW_NUMBER		ROW_NUMBER 是一种窗口函数，它返回 inline-window-specification 的当前窗口的每一行的行号。
STD	STD	返回标准差
STDDEV	STDDEV	STDDEV 从函数表达式参数求值所派生的一组数值中计算统计学标准差。这些数字

		可以加权处理。
STDDEV_POP	STDDEV_POP	STDDEV_POP 是一种窗口函数，它为 inline-window-specification 指定的当前窗口返回给定表达式所有非空值的标准差。
SUM	SUM	从函数表达式参数求值所派生的一组数值中计算总和。
	VAR_POP	总体方差
	VAR_SAMP	样本方差
VARIANCE	VARIANCE	从函数表达式参数求值所派生的一组数值中计算统计方差。这些数字可以加权处理。

3.4.8. 其它函数

QianBase	**MySQL**	**函数说明**
	ANY_VALUE	抑制 only_full_group_by 值拒绝
	BENCHMARK	重复执行表达式
AUTHNAME		返回与特定授权 ID 号相关联的授权名称。
BITAND		在两个操作数的相应位上执行'AND'运算。
CASE		一个条件表达式。两种形式的 CASE 表达式分别为 CASE 简单表达式和 CASE 搜索表达式。
	CHARSET	返回参数的字符集
	COLLATION	返回字符串参数的排序规则

	CONNECTION_ID	返回连接的连接 ID（线程 ID）
CONVERTTOHEX		将指定的值表达式转换为 16 进制数值，以便进行显示。
CURRENT_USER	CURRENT_USER	返回调用该函数的当前用户的数据库用户名。
	DEFAULT	返回表列的默认值
	FOUND_ROWS	对于带有限制子句的选择，将返回的行数没有限制子句
EXPLAIN		生成一个结果表，其描述 SELECT、INSERT、DELETE 或 UPDATE 语句的访问计划。
	GET_LOCK	获取命名锁
	INET_ATON	返回 IP 地址的数值
	INET_NTOA	从数值返回 IP 地址
	INET6_ATON	返回 IP V6 地址的数值
	INET6_NTOA	从数值返回 IP V6 地址
	LAST_INSERT_ID	最后插入的自动递增列的值
	ROW_COUNT	更新的行数
	SLEEP	延迟几秒钟
USER	USER/SESSION_USER/SYSTEM_USER	返回调用该函数的当前用户的数据库用户名或与指定的用户 ID 号相关联的数据库用户名。

	UUID()/UUID_SHORT	ReturnUUID
	VERSION	MySQL 软件版本

4. 数据库元数据迁移

4.1. Oracle 元数据迁移示例

通过 SQL Developer 可以导出表的 DDL 语句，示例如下：

```
CREATE TABLE "ZJ_USER"."TABLE1"
   (
  "COLUMN1" NUMBER NOT NULL ENABLE,
    "COLUMN2" DATE,
    "COLUMN3" VARCHAR2(20 BYTE),
    "COLUMN5" CHAR(10 BYTE),
    "COLUMN6" FLOAT(126),
    "COLUMN7" NCHAR(1),
    "COLUMN8" LONG,
    "COLUMN9" TIMESTAMP (6),
    "COLUMN4" CLOB,
     CONSTRAINT "TABLE1_PK" PRIMARY KEY ("COLUMN1")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "ZJ_USER"  ENABLE
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "ZJ_USER"
 LOB ("COLUMN4") STORE AS BASICFILE (
  TABLESPACE "ZJ_USER" ENABLE STORAGE IN ROW CHUNK 8192 RETENTION
  NOCACHE LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)) ;
```

QianBase 语句:

```
CREATE TABLE "ZJ_USER"."TABLE1"
 (
"COLUMN1" DECIMAL NOT NULL ENABLE,
  "COLUMN2" DATE,
  "COLUMN3" VARCHAR(20 BYTE),
  "COLUMN5" CHAR(10 BYTE),
  "COLUMN6" REAL,
  "COLUMN7" CHAR(1),
  "COLUMN8" VARCHAR(50),
  "COLUMN9" TIMESTAMP (6),
  "COLUMN4" CLOB,
   CONSTRAINT "TABLE1_PK" PRIMARY KEY ("COLUMN1")
 )
 ;
```

4.2. MySQL 元数据迁移示例

通过 MySQLdump -u root -p   testdb1 导出建表语句和 insert 数据语句,类似:

```
-- ------------------------------------------------------
-- Server version      5.1.73
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014      SET      @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101                SET                 @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `test`
--

DROP TABLE IF EXISTS `test`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `test` (
```

```
  `id` INT(11) DEFAULT NULL,
  `name` CHAR(10) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `test`
--

LOCK TABLES `test` WRITE;
/*!40000 ALTER TABLE `test` DISABLE KEYS */;
INSERT INTO `test` VALUES (1,'aa');
/*!40000 ALTER TABLE `test` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `test1`
--

DROP TABLE IF EXISTS `test1`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `test1` (
  `ts`  TIMESTAMP  NOT  NULL  DEFAULT  CURRENT_TIMESTAMP  ON  UPDATE
CURRENT_TIMESTAMP
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `test1`
--

LOCK TABLES `test1` WRITE;
/*!40000 ALTER TABLE `test1` DISABLE KEYS */;
INSERT INTO `test1` VALUES ('2017-02-01 16:01:03');
/*!40000 ALTER TABLE `test1` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-03-08 18:01:41
```

QianBase 支持标准的 SQL 数据操作，支持 SELECT, INSERT, UPDATE, DELETE 和 UPSERT/MERGE 语法，并提供不同方式的 JOIN、UNION、WHERE、聚合（GROUP BY 和 HAVING)、排序、取样、相关和嵌套子查询、游标和很多 SQL 函数。

通过编辑此内容，将 MySQL 的特殊语法删除，变成如下:

```
-- ------------------------------------------------------
-- Server version	5.1.73

--/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
--/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
--/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
--/*!40101 SET NAMES utf8 */;
--/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
--/*!40103 SET TIME_ZONE='+00:00' */;
--/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
--/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
--/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
--/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table test
--
```

```
DROP TABLE IF EXISTS test;
--/*!40101 SET @saved_cs_client     = @@character_set_client */;
--/*!40101 SET character_set_client = utf8 */;
CREATE TABLE test (
  Id INT DEFAULT NULL,
  name CHAR(10) DEFAULT NULL
) ;
--/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table test
--

--LOCK TABLES test WRITE;
--/*!40000 ALTER TABLE test DISABLE KEYS */;
INSERT INTO test VALUES (1,'aa');
--/*!40000 ALTER TABLE test ENABLE KEYS */;
--UNLOCK TABLES;

--
-- Table structure for table test1
--

DROP TABLE IF EXISTS test1;
--/*!40101 SET @saved_cs_client     = @@character_set_client */;
--/*!40101 SET character_set_client = utf8 */;
CREATE TABLE test1 (
  ts TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ;
```

```
--/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table test1
--

--LOCK TABLES test1 WRITE;
--/*!40000 ALTER TABLE test1 DISABLE KEYS */;
INSERT INTO test1 VALUES ('2017-02-01 16:01:03');
--/*!40000 ALTER TABLE test1 ENABLE KEYS */;
--UNLOCK TABLES;
--/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

--/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
--/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
--/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
--/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
--/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
--/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
--/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-03-08 18:01:41
```

修改对比：

	MySQL	QianBase
注释/*	/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */	--/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */

存储引擎	ENGINE=MyISAM DEFAULT CHARSET=latin1	
UNLOCK 表	UNLOCK TABLES	--UNLOCK TABLES;
LOCK 表	LOCK TABLES test1 WRITE;	--LOCK TABLES test1 WRITE;
表名和列名定义	\`test1\` \`name\`	test1 name
INT 整形	Int(11)	Int
时间戳定义	ON UPDATE CURRENT_TIMESTAMP	

4.3. 支持保留关键字

在迁移过程中，DDL 或者 DML 可能包含一些 QianBase 的保留关键字，如果它们是 SQL 的一部分， SQL 将这些文字视为保留字。它们不得被用作标识符，除非您用双引号将其括起来。保留关键字参见附录 A 内容;

比如:

源 DDL 语句	**目标 DDL 语句**
CREATE TABLE TABLE1 (ID NUMBER NOT NULL, RELATION_KEY VARCHAR2(50) NOT NULL, NAME VARCHAR2(50) NOT NULL, TABLESNAME VARCHAR2(50) NOT NULL, CONNECTNAME VARCHAR2(50) NOT NULL, SQL VARCHAR2(255) NOT NULL, ,PRIMARY KEY(id));	CREATE TABLE TABLE1 (ID decimal NOT NULL, RELATION_KEY VARCHAR2(50) NOT NULL, NAME VARCHAR2(50) NOT NULL, TABLESNAME VARCHAR2(50) NOT NULL, CONNECTNAME VARCHAR2(50) NOT NULL, "SQL" VARCHAR2(255) NOT NULL, ,PRIMARY KEY(id));

5. 数据迁移

5.1. DML 方式迁移数据

Oracle 数据导出方式:

方式	适合场景	描述
Oracle Export	Oracle 数据库之间迁移	导出为.dmp 的文件格式，.dmp 文件是二进制的，可以跨平台，还能包含权限，效率也很不错，用得最广
SQL Insert	适合异构数据库迁移	导出为.sql 文件，可用文本编辑器查看，通用性比较好，但效率不如第一种，适合小数据量导入导出。尤其注意的是表中不能有大字段（CLOB BLOB），如果有，会提示不能导出，可以用第一种和第三种方式导出。
PL/SQL developer	Oracle 数据库之间迁移	导出为.pde 格式的，.pde 为 Pl/sql developer 自有的文件格式，只能用 Pl/sql developer 自己导入导出；不能用编辑器查看
CSV/TXT 文件	适合异构数据库迁移	通过 SQL Developer 导出文件文件，CSV 文件以逗号分隔列值，TXT 文件以空格分隔列值

MySQL 通过 MySQLdump，Oracle 可以通过 SQL Developer 工具从原数据库导出 INSERT 语句到文件，在目标数据库直接运行脚本文件;

适合场景：数据量不大，比较适合;

步骤:

a) 源数据库:从原数据库 Oracle、MySQL 导出 INSERT 语句到 insert-traf.sql 文件;

比如上表导出 Oracle 语句:

```
REM INSERTING into TABLE1
INSERT INTO TABLE1
(COLUMN1,COLUMN2,COLUMN3,COLUMN5,COLUMN6,COLUMN7,COLUMN8,COLUMN9)
VALUES(111,to_timestamp('09-3 月 -17','DD-MON-RR HH.MI.SS.FF
AM'),'1111','111 ',111,'1','111',to_timestamp('09-3 月 -17
12.00.00.000000000 上午','DD-MON-RR HH.MI.SS.FF AM'));
```

b) 目标数据库:

修改 INSERT 文件,直接在目标数据库运行 INSERT 语句脚本;

c) 运行命令:

```
trafci.sh -h 10.1.1.1:23400 -u zj_user -p zj_user -s insert-traf.sql
```

5.2. ODB 迁移数据

5.2.1. ODB 介绍

ODB 是易鲸捷公司开发的专门用于 QianBase trickle load 的工具,它使用了最佳的数据加载优化方式,并支持所有主流的数据源和目标数据库,可以满足多数数据迁移需求。通过 ODB ODBC 工具,直接实现数据的导入和导出,中间数据不落地;

Trickle load 对数据进行实时的加载,即在数据到达的同时即加载进入数据库,但数据量不大;

ODB 是一个基于 ODBC 的数据迁移工具。它支持多种数据源，可以作为数据的 loader，也可以作为 extractor。ODB 支持 Linux 和 Windows。ODB 综合使用了以下三种技术。

- 使用Rowset批量插入
- 使用UPSERT命令
- 使用并发连接，同时导入数据

5.2.2. 安装和配置

安装 ODBC 驱动程序，比如源数据库（Oracle，MySQL）和目标 QianBase 的 ODBC 驱动程序，配置 ODBC 驱动程序和 DSN

数据库类型	驱动程序
MYSQL	MySQL-connector-odbc-5.3.6-1.el6.x86_64.rpm
Oracle	instantclient-odbc-linux.x64-11.2.0.4.0.zip
QianBase	TRAF_ODBC_Linux_Driver_64.tar.gz

数据库类型	odbcinst.ini配置
MYSQL	[MySQL] Description = MySQL driver Driver = /usr/lib/libmyodbc5.so Setup = /usr/lib/libodbcmyS.so Socket = /tmp/MySQL.sock FileUsage = 1
Oracle	[Oracle 11g ODBC driver] Description = Oracle ODBC driver for Oracle 11g

	Driver = /home/oracle_driver/instantclient_11_2/libsqora.so.11.1 Setup = FileUsage = CPTimeout = CPReuse =
QianBase	[Trafodion] Description = Trafodion ODBC Stand Alone Driver Driver = /usr/lib64/libtrafodbc_drvr64.so FileUsage = 1 UsageCount = 1 AppUnicodeType = utf16

数据库类型	odbc.ini配置
MYSQL	[MySQL-test] Description = MySQL test database Trace = On TraceFile = stderr Driver = MySQL SERVER = localhost USER = root PASSWORD = kaychen PORT = 3306 DATABASE = test socket = /tmp/MySQL.sock
Oracle	[orac] driver = Oracle 11g ODBC driver server = 192.168.1.1 port = 1522

	ServerName = XE
QianBase	[traf] Description = traf instance in guiyang 10.10.11.3 Driver = Trafodion Catalog = TRAFODION Schema = SEABASE DataLang = 0 FetchBufferSize = SYSTEM_DEFAULT Server = TCP:192.168.1.27:23400 SQL_ATTR_CONNECTION_TIMEOUT = SYSTEM_DEFAULT SQL_LOGIN_TIMEOUT = SYSTEM_DEFAULT SQL_QUERY_TIMEOUT = NO_TIMEOUT ServiceName = HP_DEFAULT_SERVICE

5.2.3. 数据迁移示例

Oracle 迁移使用示例:

```
odb64luo -u src_user:tgt_user -p src_pwd:tgt:pwd -d orac:traf
    -cp src=table1:tgt=table1
```

MySQL 迁移使用示例:

```
odb64luo -u src_user:tgt_user -p src_pwd:tgt:pwd
    -d MySQL-test:traf -cp src=table1:tgt=table1
```

5.3. ETL 工具迁移数据

QianBase 可以进行简单的数据清洗，更加复杂的数据清洗工作则需要利用 QianBase 的 JDBC/ODBC 接口开发专门的程序或使用专业的 ETL 软件，比如 Penhato Data Integration（Kettle）等工具;

Penhato Data Integration 可以实现不同数据库之间的数据同步，通过表输入和表输出 2 个转换即可实现不同数据库之间的表数据同步；

限制：

1. 目前 Kettle 不支持 BLOB、CLOB 大对象数据迁移；

5.4. Sqoop+Bulkload 迁移数据

5.4.1. 迁移过程

Bulk load 使用 QianBase 的 bulkloader 工具和命令进行。

可以使用开源的 scoop 或者源数据库自带的数据迁移工具将数据迁移到 HDFS/HIVE 中;然后利用 QianBase 的 Bulkloader 进行高速的数据导入;

这种方法适用于海量数据的加载。一般是对数据仓库或者数据湖泊最初的初始化导入。

5.4.2. Sqoop 数据导出

Sqoop 导出数据到 HIVE:

前提：拷贝对应源数据库的 JDBC 驱动程序 jar 文件到 sqoop/lib 目录下；

1. 表没有 BLOB，CLOB

可以直接把数据迁移到 HIVE，HIVE 自动创建表

```
sqoop import --hive-import \
```

```
--connect jdbc:oracle:thin:@192.168.1.1:1521:orcl \
--username zj_user --password zj_user --verbose -m 1 \
--table ADMIN
```

2. 表有 CLOB
3. CLOB 可以直接导入到 HIVE

```
    sqoop import \
    -Dmapred.job.queue.name=default \
    --connect jdbc:oracle:thin:@192.168.1.1:1521:orcl \
--username zj_user --password zj_user \
    --table TABLE1 --fields-terminated-by '|' -m 1 \
    --target-dir /bulkload/zj_user/TABLE1 \
    -hive-drop-import-delims \
    -map-column-java CL1=String, CL2=String \
    -m 1 \
    -hive-import \
    -hive-table TABLE1 \
    -split-by CL3
```

4. BLOB+CLOB

BLOB 不能直接导入到 HIVE，可以导入到 HDFS

```
    sqoop import \
    -Dmapred.job.queue.name=default \
    --connect jdbc:oracle:thin:@192.168.1.1:1521:orcl \
--username zj_user --password zj_user \
    --table TABLE1 \
    --target-dir /bulkload/zj_user/TABLE11 \
    -inline-lob-limit 16777216 \
    -m 2
```

5.4.3. Bulkload 数据导入

Bulkload 导入数据到 QianBase：

预先在 QianBase 创建对应的 ADMIN 表

Trafci

load into admin SELECT * FROM hive.hive.admin

5.5. 文本文件+BulkLoad 方式

5.5.1. 文件数据导出

数据库	文件文件	参数
MySQL	使用 SELECT INTO outfile "filename"语句	可以指定卸出文件时，字段之间的分隔字符，转义字符，包括字符，及记录行分隔字符。列在下面： FIELDS TERMINATED BY "\t" [OPTIONALLY] ENCLOSED BY "" ESCAPED BY "\\" LINES TERMINATED BY "\n"
MySQL	MySQLdump -T./ testdb	指定字段分隔符，包括字符，转义字段，行记录分隔符。参数列在下面： --fields-terminated-by= 字段分隔符 --fields-enclosed-by= 字段包括符 --fields-optionally-enclosed-by=

		字段包括符，只用在 CHAR 和 VERCHAR 字段上 --fields-escaped-by= 转义字符 --lines-terminated-by= 行记录分隔符
Oracle 数据库	通过SQL Developer导出文件文件	CSV 文件以逗号分隔列值，TXT 文件以空格分隔列值;

5.5.2. 数据上传到 HDFS

```
hadoop fs -mkdir /bulkload/customer_demographics
hadoop fs -copyFromLocal customer_demographics
            /bulkload/customer_demographics
```

5.5.3. 创建 hive 外部表

hive→

```
CREATE EXTERNAL TABLE customer_demographics
(
cd_demo_sk INT
, cd_gender STRING
, cd_marital_status STRING
, cd_education_status STRING
, cd_purchase_estimate INT
, cd_credit_rating STRING
, cd_dep_count INT
, cd_dep_employed_count INT
, cd_dep_college_count INT
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' LOCATION
'/bulkload/customer_demographics';

SELECT COUNT(*) FROM customer_demographics;
```

5.5.4. Load 数据到 QianBase 表

trafci→

```
SELECT * FROM hive.hive.customer_demographics LIMIT 5;

CREATE TABLE customer_demographics_hbase
(
cd_demo_sk INT NOT NULL
, cd_gender CHAR(1)
, cd_marital_status CHAR(1)
, cd_education_status CHAR(20)
, cd_purchase_estimate INT
, cd_credit_rating CHAR(10)
, cd_dep_count INT
, cd_dep_employed_count INT
, cd_dep_college_count INT
, PRIMARY KEY (cd_demo_sk)
)
salt using 6 partitions on (cd_demo_sk);

load into customer_demographics_hbase SELECT * FROM
hive.hive.customer_demographics WHERE cd_demo_sk <= 5000;
```

6. 应用迁移

6.1. 分页功能

MySQL 分页功能：LIMIT {[offset,] row_count | row_count OFFSET offset}

Oracle 分页功能：通过 ROWNUM 实现，

比如：SELECT * FROM

(

SELECT A.*, ROWNUM RN

FROM (SELECT * FROM TABLE_NAME) A

WHERE ROWNUM <= 40

```
)
WHERE RN >= 21
```

QianBase实现:兼容OracleROWNUM分页方式,同时也可以通过row_number窗口函数实现

```
SELECT * FROM
    (SELECT row_number() over(order by ID) rownumber,* FROM
    (SELECT * FROM T_1))
WHERE    rownumber < 20 AND rownumber > 10;
```

6.2. Java 应用迁移

6.2.1. JDBC 驱动

QianBase 提供标准的 JDBC 驱动程序，实现 JDBC 功能，

jar 文件：$TRAF_HOME/export/lib/jdbcT4.jar

6.2.2. JAVA 程序写法

1. 数据库信息:

    IP 地址是：192.168.1.1

    端口：23400

    用户名：trafodion，密码：traf123

2. 程序声明

```
import org.trafodion.jdbc.t4.*;
```

```
private final static String URL = "jdbc:t4jdbc://192.168.1.1:23400/:";
private final static String DRIVER_CLASS = "org.trafodion.jdbc.t4.T4Driver";
private final static String USERNAME = "trafodion";
private final static String PASSWORD = "traf123";
```

3. DriverManager 连接方法

```
System.out.println("-----DriverManager connection Test-----------");
Connection mConn;
Class.forName(DRIVER_CLASS);
mConn = DriverManager.getConnection(URL, USERNAME, PASSWORD);
```

4. DataSource 连接方法

```
System.out.println("-----DataSource connection Test-----------");
HPT4DataSource ds = new HPT4DataSource() ;
ds.setUrl( URL );
ds.setCatalog( "TRAFODION" ) ;
ds.setSchema( "SEABASE" ) ;
ds.setUser( USERNAME ) ;
ds.setPassword( PASSWORD ) ;
ds.setMaxPoolSize( "30" ) ;
ds.setMinPoolSize( "10" ) ;
mConn = ds.getConnection() ;
```

6.3. Hibernate 配置

通常情况下，有特殊语法才需要方言

缺省方言：org.hibernate.dialect.Dialect

applicationContext-hibernate.xml 里配置

```
<bean id="dataSource_trafodion"
class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
        <property name="driverClassName">
            <value>org.trafodion.jdbc.t4.T4Driver</value>
        </property>
        <property name="url">
            <value>jdbc:t4jdbc://10.9.39.150:23400/schema=solar</value>
        </property>
        <property name="username">
        <value>trafodion</value>
    </property>
    <property name="password">
        <value>trafodion</value>
    </property>
    </bean>

    <!-- Hibernate //-->
    <bean id="sessionFactory"
        class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="dataSource">
        <ref bean="dataSource_trafodion" />
    </property>
        <!--<property name="configLocation"
value="classpath:hibernate.cfg.xml">
     </property> -->
    <property name="hibernateProperties">
```

```
        <props>
            <!--<prop
        key="hibernate.dialect">org.hibernate.dialect.Oracle9Dialect</prop> -->
            <prop key="hibernate.dialect">${db.hibernate.dialect}</prop>
            <prop key="jdbc.batch_size">1024</prop>
            <prop key="hibernate.show_sql">false</prop>
            <prop key="hibernate.cglib.use_reflection_optimizer">true</prop>
        </props>
    </property>
```

附录 A--保留 SQL 标识符

在迁移过程中，DDL 或者 DML 可能包含一些保留关键字，如果它们是 SQL 的一部分， SQL 将这些文字视为保留字。它们不得被用作标识符，除非您用**双引号**将其括起来。

保留的SQL标识符-A

ACTION	ADD	ADMIN	AFTER	AGGREGATE
ALIAS	ALL	ALLOCATE	ALTER	AND
ANY	ARE	ARRAY	AS	ASC
ASSERTION	ASYNC	AT	AUTHORIZATION	AVG

保留的SQL标识符-B

BEFORE	BEGIN	BETWEEN	BINARY	BIT
BIT_LENGTH	BLOB	BOOLEAN	BOTH	BREADTH
BY				

保留的SQL标识符-C

CALL	CASCADE	CASCADED	CASE	CAST
CATALOG	CHAR	CHARACTER	CHARACTER_LENGTH	CHAR_LENGTH
CHECK	CLASS	CLOB	CLOSE	COALESCE
COLLATE	COLLATION	COLUMN	COMMIT	COMPLETION
CONNECT	CONNECTION	CONSTRAINT	CONSTRAINTS	CONSTRUCTOR
CONTINUE	CONVERT	CORRESPONDING	COUNT	CREATE
CROSS	CUBE	CURRENT	CURRENT_DATE	CURRENT_PATH
CURRENT_R	CURRENT_TI	CURRENT_TIMEST	CURRENT_USER	CURRNT_USR_IN

OLE	ME	AMP		TN
CURSOR	CYCLE			

保留的SQL标识符-D

DATE	DATETIME	DAY	DEALLOCATE	DEC
DECIMAL	DECLARE	DEFAULT	DEFERRABLE	DEFERRED
DELETE	DEPTH	DEREF	DESC	DESCRIBE
DESCRIPTOR	DESTROY	DESTRUCTOR	DETERMINISTIC	DIAGNOSTICS
DICTIONARY	DISCONNECT	DISTINCT	DOMAIN	DOUBLE
DROP	DYNAMIC			

保留的SQL标识符-E

EACH	ELSE	ELSEIF	END	END-EXEC
EQUAL	ESCAPE	EXCEPT	EXCEPTION	EXEC
EXECUTE	EXISTS	EXTERNAL	EXTRACT	

保留的SQL标识符-F

FALSE	FETCH	FIRST	FLOAT	FOR
FOREIGN	FOUND	FRACTION	FREE	FROM
FULL	FUNCTION			

保留的SQL标识符-G

GENERAL	GET	GLOBAL	GO	GOTO
GRANT	GROUP	GROUPING		

保留的SQL标识符-H

HAVING	HOST	HOUR		

保留的SQL标识符-I

IDENTITY	IF	IGNORE	IMMEDIATE	IN
INDICATOR	INITIALLY	INNER	INOUT	INPUT
INSENSITIVE	INSERT	INT	INTEGER	INTERSECT
INTERVAL	INTO	IS	ISOLATION	ITERATE

保留的SQL标识符-J

JOIN

保留的SQL标识符-K

KEY

保留的SQL标识符-L

LANGUAGE	LARGE	LAST	LATERAL	LEADING
LEAVE	LEFT	LESS	LEVEL	LIKE
LIMIT	LOCAL	LOCALTIME	LOCALTIMESTAMP	LOCATOR
LOOP	LOWER			

保留的SQL标识符-M

MAINTAIN	MAP	MATCH	MATCHED	MAX
MERGE	MIN	MINUTE	MODIFIES	MODIFY
MODULE	MONTH			

保留的SQL标识符-N

NAMES	NATIONAL	NATURAL	NCHAR	NCLOB
NEW	NEXT	NO	NONE	NOT
NULL	NULLIF	NUMERIC		

保留的SQL标识符-O

OCTET_LENGTH	OF	OFF	OID	OLD
ON	ONLY	OPEN	OPERATORS	OPTION
OPTIONS	OR	ORDER	ORDINALITY	OTHERS
OUT	OUTER	OUTPUT	OVERLAPS	

保留的SQL标识符-P

PAD	PARAMETER	PARAMETERS	PARTIAL	PENDANT
POSITION	POSTFIX	PRECISION	PREFIX	PREORDER
PREPARE	PRESERVE	PRIMARY	PRIOR	PRIVATE
PRIVILEGES	PROCEDURE	PROTECTED	PROTOTYPE	PUBLIC

保留的SQL标识符-Q

QUALIFY

保留的SQL标识符-R

READ	READS	REAL	RECURSIVE	REF
REFERENCES	REFERENCING	RELATIVE	REORG	REORGANIZE
REPLACE	RESIGNAL	RESTRICT	RESULT	RETURN
RETURNS	REVOKE	RIGHT	ROLLBACK	ROLLUP

ROUTINE	ROW	ROWS	PROTOTYPE	

保留的SQL标识符-S

SAVEPOINT	SCHEMA	SCOPE	SCROLL	SEARCH
SECOND	SECTION	SELECT	SENSITIVE	SESSION
SESSION_USER	SESSN_USR_INTN	SET	SETS	SIGNAL
SIMILAR	SIZE	SMALLINT	SOME	SPECIFIC
SPECIFICTYPE	SQL	SQL_CHAR	SQL_DATE	SQL_DECIMAL
SQL_DOUBLE	SQL_FLOAT	SQL_INT	SQL_INTEGER	SQL_REAL
SQL_SMALLINT	SQL_TIME	SQL_TIMESTAMP	SQL_VARCHAR	SQLCODE
SQLERROR	SQLEXCEPTION	SQLSTATE	SQLWARNING	STRUCTURE
SUBSTRING	SUM	SYNONYM	SYSTEM_USER	

保留的SQL标识符-T

TABLE	TEMPORARY	TERMINATE	TEST	THAN
THEN	THERE	TIME	TIMESTAMP	TIMEZONE_HOUR
TIMEZONE_MINUTE	TO	TRAILING	TRANSACTION	TRANSLATE
TRANSLATION	TRANSPOSE	TREAT	TRIGGER	TRIM
TRUE				

保留的SQL标识符-U

UNDER	UNION	UNIQUE	UNKNOWN	UNNEST
UPDATE	UPPER	UPSHIFT	USAGE	USER

USING				

保留的SQL标识符-V				
VALUE	VALUES	VARCHAR	VARIABLE	VARYING
VIEW	VIRTUAL	VISIBLE		

保留的SQL标识符-W				
WAIT	WHEN	WHENEVER	WHERE	WHILE
WITH	WITHOUT	WORK	WRITE	

保留的SQL标识符-Y
YEAR

保留的SQL标识符-Z
ZONE