

# Sakai 11 database support

The Sakai CLE supports the following production-grade databases:

Database	Driver	Notes
MySQL 5.5.x	<a href="#">MySQL Connector/J 5.1.13+</a>	<p>⊖ Sakai requires transaction support. In the case of MySQL you must implement the <code>InnoDB</code> storage engine to ensure proper transaction handling.</p> <p>★ QA for Sakai 10 MySQL was using Connector/J 5.1.27 and MySQL 5.5.34</p>
MySQL 5.1.x	<a href="#">MySQL Connector/J 5.1.13+</a>	<p>⊖ Sakai requires transaction support. In the case of MySQL you must implement the <code>InnoDB</code> storage engine to ensure proper transaction handling.</p>
Oracle 11g	OJDBC	<p>⚠ Oracle recommends using the latest 11g driver if you are using Java 6 with either 10g or 11g; with Java 6 in combination with the 11g driver,</p> <p>★ Recommend but not required - convert longs to clobs -</p> <p>☑ <a href="#">SAK-7977</a> - Change fields from long to clob for Oracle <span style="border: 1px solid gray; padding: 2px;">CLOSED</span> Only tested on Oracle 11.2.0.3 - <a href="https://github.com/daveadams/sakai-tools/blob/master/convert-longs-to-lobbs.sql">https://github.com/daveadams/sakai-tools/blob/master/convert-longs-to-lobbs.sql</a> . By default it will scan your tables for LONG and LONG RAW fields but it won't change anything. Add "CONVERT" to the command line to actually perform the conversion and rebuild the indexes in the involved tables. Please be sure you understand what the script will be doing before you run it, and always always test thoroughly on a non-production environment before you attempt to run this on your production system.</p>
Oracle 10g	OJDBC	<p>⚠ Both Oracle 10g AND 9i users must use the 10g driver; the latest 10g "Release 2" (10.2.x) or higher is recommended.</p> <p>⚠ Oracle recommends using the latest 11g driver if you are using Java 6 with either 10g or 11g; with Java 6 in combination with the 11g driver</p>
DBZ		Deprecated

The Sakai CLE have official support for Mysql with production-grade databases, but MariaDB is a binary drop in replacement for Mysql (source : <https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-compatibility/>) and should work with Sakai but it is not officially supported.

Database	Driver	Notes
MariaDB 5.5.x (Not officially supported but works)	<a href="#">MySQL Connector/J 5.1.13+</a>	<p>⊖ Sakai requires transaction support. In the case of MariaDB, you must implement the <code>XtraDB</code> storage engine to ensure proper transaction handling.</p> <p>Check the Incompatibilities between MariaDB 5.5 and MariaDB 5.3 and MySQL 5.5 to configure your MariaDB server correctly if you want to use the "INSERT IGNORE" parameter:</p> <p><a href="https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-compatibility/#incompatibilities-between-mariadb-55-and-mariadb-53-and-mysql-55">https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-compatibility/#incompatibilities-between-mariadb-55-and-mariadb-53-and-mysql-55</a></p> <p>Rutgers University, The State University of New Jersey, use Sakai 2.9.x with MariaDB 5.5 and the XtraDB storage engine (previously MariaDB 5.1 and Sakai 2.9.x)</p> <p>Source 1 : <a href="https://sakai.rutgers.edu/access/content/public/technical.html">https://sakai.rutgers.edu/access/content/public/technical.html</a></p> <p>Source 2 : <a href="http://collab.sakaiproject.org/pipermail/production/2013-January/001962.html">http://collab.sakaiproject.org/pipermail/production/2013-January/001962.html</a></p>

Choose the appropriate MySQL or Oracle JDBC driver (or connector) for your installation. For MySQL, download the `*zip/*tar.gz` archive, extract its contents and copy the `mysql-connector-java-<version>-bin.jar` to `$CATALINA_HOME/common/lib`. For Oracle download the `ojdbc14.jar` file and copy it to `$CATALINA_HOME/common/lib`.

📘 You will **not** need to create Sakai database objects (tables, indices, etc) when setting up your database. Sakai generates its own database schema automatically during the Tomcat setup process via the `autoDDL` setting in `sakai.properties`.

In actuality, Sakai is not limited to these database choices and integration with other RDBMS systems is not difficult. In the past at least one installation used Microsoft SQL Server while requests for PostgreSQL integration are occasionally raised on the Sakai developers list. However, to date no one in the Sakai Community has stepped forward to support alternatives to either Oracle or MySQL, a prerequisite for adding additional database options to the release.

⊖ Irrespective of whether you utilize MySQL or Oracle, be sure you have configured your database to use the UTF-8 character set. Failure to do so will result in range of issues when attempting to use Unicode characters in Sakai. Consult your database documentation or a local DBA for instructions on how to set your database up properly.

If you are uncertain as to how your database is currently configured, you can check with a query. Here is a sample query for checking an Oracle instance:

```
SQL> select value from nls_database_parameters where parameter = 'NLS_CHARACTERSET';

VALUE
-----
AL32UTF8
```

For MySQL, the command to see what encoding your database is currently set to is "show create database sakai", assuming, of course, that your database is named "sakai". e.g.:

```
mysql> show create database sakai;
+-----+
| Database | Create Database |
+-----+
| sakai | CREATE DATABASE `sakai` /*!40100 DEFAULT CHARACTER SET utf8 */ |
+-----+
1 row in set (0.00 sec)
```

Converting a database from one character set to another is a non-trivial operation, particularly if it is a large production database. We recommend strongly that you verify this aspect of the database creation and configuration process before deploying Sakai.

## 1.0 Configure the database

By default, all Sakai distributions are configured to use an in-memory version of HSQLDB. HSQLDB is provided for testing/demo purposes only and should not be run in production.

Whatever database you choose to use you will need to modify at a minimum the following connection settings in `sakai.properties`:

```
url@javax.sql.DataSource
username@javax.sql.DataSource
password@javax.sql.DataSource
```

## 2.0 Set the database username and password

Set your database username and password:

```
# DATABASE CONFIGURATION - make sure to modify details to match your particular setup

# The username and password. The defaults are for the out-of-the-box HSQLDB. Change to match your setup.
username@javax.sql.DataSource=yourDbUserName
password@javax.sql.DataSource=yourDbPassword
```

## 3.0 Set the database connection

### MySQL Sample Configuration

Locate the MySQL configuration block, uncomment the settings and save your changes. Make sure you modify the data source, username and password settings to match your local environment. Do not forget to comment out the HSQLDB and Oracle settings.

```
# MySQL settings - make sure to alter as appropriate
vendor@org.sakaiproject.db.api.SqlService=mysql
driverClassName@javax.sql.DataSource=com.mysql.jdbc.Driver
hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
url@javax.sql.DataSource=jdbc:mysql://127.0.0.1:3306/sakai?useUnicode=true&characterEncoding=UTF-8
validationQuery@javax.sql.DataSource=
defaultTransactionIsolationString@javax.sql.DataSource=TRANSACTION_READ_COMMITTED
```

## Oracle Sample Configuration

Locate the Oracle configuration block, uncomment the settings and save your changes. Make sure you modify the data source, username and password settings to match your local environment. Do not forget to comment out the HSQLDB and MySQL settings.

```
# Oracle settings - make sure to alter as appropriate
vendor@org.sakaiproject.db.api.SqlService=oracle
driverClassName@javax.sql.DataSource=oracle.jdbc.driver.OracleDriver
#hibernate.dialect=org.hibernate.dialect.Oracle9iDialect
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
url@javax.sql.DataSource=jdbc:oracle:thin:@your.oracle.dns:1521:SID
validationQuery@javax.sql.DataSource=select 1 from DUAL
defaultTransactionIsolationString@javax.sql.DataSource=TRANSACTION_READ_COMMITTED
```

⚠ Oracle users may experience performance issues with certain of the SQL settings that work for MySQL. Oracle users can reduce Db load by uncommenting the following settings:

```
# For improved Oracle performance (from the University of Michigan)
#validationQuery@javax.sql.DataSource=
#defaultTransactionIsolationString@javax.sql.DataSource=
#testOnBorrow@javax.sql.DataSource=false
```

## 4.0 Set `auto.ddl=true`

ℹ On startup, Sakai will generate all database objects (tables, keys, constraints, etc.) automatically, obviating the need to run DDL scripts manually per the `sakai.properties` setting `auto.ddl`.

```
# establish auto.ddl - on by default
auto.ddl=true
#auto.ddl=false
```

⚠ Once the database schema is created you should set `auto.ddl=false`.

## 5.0 Install MySQL 5.5 / 5.1

Configuring the Sakai CLE to use MySQL is an excellent option both for local development and production purposes.

⊖ Sakai **requires transaction support**. In the case of MySQL you must implement the `InnoDB` storage engine to ensure proper transaction handling.

⚠ Users upgrading from MySQL 5.0 to MySQL 5.1 on a Mac have reported name comparison problems on startup (for more info see <http://collab.sakaiproject.org/pipermail/sakai-dev/2010-June/008066.html>). This issue is solved by specifying the following property in a local `my.cnf` configuration file:

```
lower_case_table_names=1
```

ℹ You will **not** need to create Sakai database objects (tables, indices, etc) when setting up your database. Sakai generates its own database schema automatically during the Tomcat setup process via the `autoDDL` setting in `sakai.properties`.

**i** Case sensitivity. In most varieties of UNIX, MySQL is case sensitive since the underlying operating system determines the case sensitivity of database and table names. This is not the case for MySQL running in Windows which is case insensitive. Previous MySQL conversion scripts contained a mixture of upper case and lower case statements which occasionally caused problems for upgraders; for 2.8 we opted for lower case for the syntax while leaving the table and field names upper case. Testing on MySQL 5.1 did not reveal any issues with this choice.

Before installing MySQL, confirm whether or not it is already installed on your system by checking the version from the command line:

```
mysql --version
mysql Ver 14.14 Distrib 5.1.52, for apple-darwin10.3.0 (i386) using readline 5.1
```

If MySQL is not installed download MySQL 5.1/5.5 binaries or source from <http://dev.mysql.com/downloads/>. Linux users should install MySQL using a package or binaries if possible. Choose the standard configuration. Windows users should consider installing MySQL as a service. Remember to include MySQL's /bin directory in your PATH statement.

MySQL 5.5: <http://dev.mysql.com/downloads/mysql/5.5.html>

MySQL 5.1: <http://dev.mysql.com/downloads/mysql/5.1.html>

Assign a password for the root account:

```
mysql -u root -pmysqlpassword
```

## 5.1 Set MySQL environment variables

### MYSQL\_HOME

Set the MYSQL\_HOME environment variable to point to the base directory of your MySQL installation. In the example below, the path points to a symbolic link file rather than the actual MySQL installation directory. You can create a symbolic link ("sym link") in Unix using the `ln -s` command (e.g., `ln -s mysql-5.1.46-osx10.6-x86_64 mysql`).

Unix/Mac:

```
export MYSQL_HOME=/usr/local/mysql
```

Windows:

```
set MYSQL_HOME=C:\mysql
```

### PATH

Add MySQL's /bin directory to your PATH variable.

Unix/Mac:

```
export PATH=$PATH:$CATALINA_HOME/bin:$JAVA_HOME/bin:$M2_HOME/bin:$MYSQL_HOME/bin:$PHP_HOME/bin:$SUBVERSION_HOME/bin:$USR_LOCAL/bin
```

Windows:

```
;C:\mysql\bin
```

## 5.2 Create a my.cnf/my.ini configuration file

You can configure MySQL to read a wide variety of startup options from a file named `my.cnf`. Settings are scoped and can be overridden by file location:

Scope	Location	Override
global	/etc/my.cnf	
server-specific	binary install: /usr/local/mysql/data/my.cnf; source install: /usr/local/var/my.cnf	global
user	~/my.cnf	global, server-specific

Below is a minimalist MySQL 5.1 configuration file for a development laptop `/etc/my.cnf`:

```
[mysqld]
default-storage-engine=InnoDB
lower_case_table_names=1
```

⚠ Restart MySQL in order for the new settings to take effect.

### 5.3 Create a "sakai" database and user account

Unable to render {include} The included page could not be found.