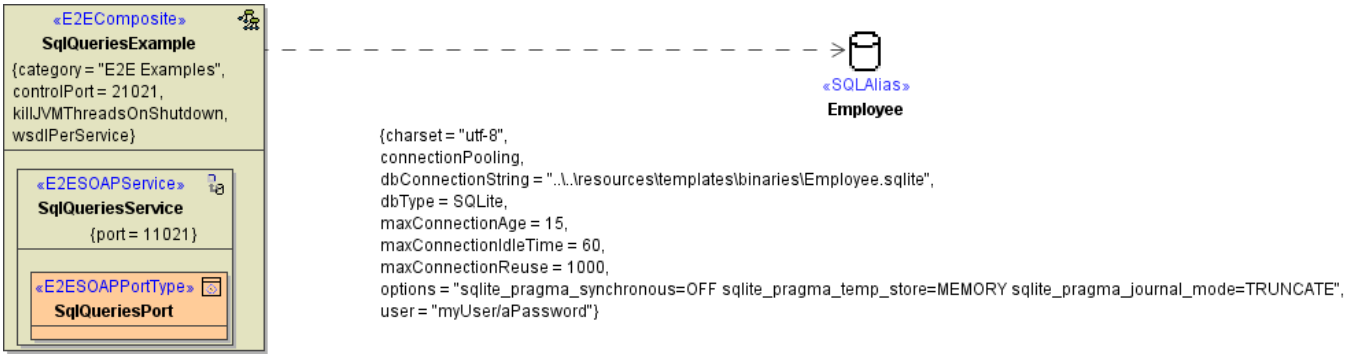
SQL Deployment

> This page explains the **SQL Adapter** in Bridge context. If you were looking for the same information regarding the PAS Designer, refer to SQL Adapter in the Designer guide.

The following figure shows a component diagram that features an SQLLite database. The database type and name are given as tagged values (**dbType** and **dbConnectionString**). Refer to Database Server-Specific Notes for SQL Adapters for more details.

Figure: SQL Component Diagram Showing Connection Options



The <<SQLAlias>> **Employee** connects the xUML service with the sql database. Each <<SQLAdapter>> referencing the same alias connects to the same database. The options of these database connections are given as tagged values on the alias :

Tagged Value	Description	Allowed Values
General		
dbConnectionString	The format of the database connection string depends on the type of the database. For more details see Database Server-Specific Notes for SQL Adapters.	
dbType	Type of the database.	Oracle, SQLServer, InterBase, SQLBase, ODBC, DB2, Informix, Sybase, MySQL, PostgreSQL, SQLite, DBTypeVariable
dbTypeVariable	If the tagged value **dbType** is set to **DBTypeValue**, the **dbTypeVariable** tagged value is used to define the type of the database. The **dbType** then can be defined by a setting variable. This is to handle the case, that you not want to hard code the **dbType**, but to configure it at runtime via the E2E Bridge. See Using Global Setting Variables for more information on how to define a global setting variable in the E2E Bridge. Use one of the listed **dbTypes** in the settings variable. If you configure an unknown **dbType** via the E2E Bridge, the xUML Runtime will throw an error on service startup.	Any global setting variable from the E2E Bridge. Example: `{{my_setting_variable}}`
user	DB user. Optional the password can be given after a '/'. However, this is recommended for development purposes only.	Example: `{{DB_USER}}/{{DB_PASSWORD}}`
options	This tagged value can hold a comma separated list of <name>=<value> pairs. These list elements are interpreted as native options. The possible name-value pairs depend on the database type. A comprehensive list can be found at https://www.sqlapi.com/ApiDoc/servers/	Example: `SSPROP_INIT_ENCRYPT=VARIANT_TRUE`
transactionIsolationLevel	Bridge 7 Specify here the required transaction isolation level of the SQL connection according to SQL-92 standard. Refer to Wikipedia for a detailed description of the available isolation levels. Please note that not all databases support all levels. In this case a database-specific mapping will occur. For persistent state databases no other than **<UNSPECIFIED>** and **DBMS default** are allowed.	DBMS default: Use the default isolation level of the connected database system. Read uncommitted: Lowest isolation level. Dirty reads allowed, SQL adapter may fetch not-yet-committed changes of other transactions. Read committed: Lock-based concurrency control.

		Repeatable read	Lock-based concurrency control.
		Serializable	Highest isolation level. Lock-based concurrency control.
Localization			
charset	Any database uses a charset to encode Strings. If the database uses UNICODE charsets (UTF-8, UTF-16, UTF-32), encoding is handled automatically. If the database is not UNICODE compliant, the Bridge assumes 7-bit ASCII by default. However, in many cases it necessary to define the charset explicitly. This is done by the tagged value **charset** as shown below. The charset needs to be the same as defined at the database settings. All possible charset definitions are listed in section Charset Definitions.	Example: `UTF-8` See Charset Definitions for a list of possible values.	
timezone	You can enter a valid time zone or the value **local**, which uses the time zone of the xUML service. See Time Zones for a list of possible values. If **timezone** does not contain any content (is NULL), UTC is used.	**Default is NULL**	Example: `"Australia/Melbourne", "CET", "Etc/GMT+10"`
unicodeMode	Added in Builder 6.0.15.5 Runtime 2015.15 Specify the encoding for database access. We recommend to use the **Platform default** unless you suspect an encoding incompatibility (see Troubleshooting the SQL Adapter). This option represents the former behavior and is fully backwards-compatible - means, it can be used with older xUML Runtimes. The two other (force mode) options will be ignored by older Runtimes without warning.	**Platform default** (default)	Use the platform default mode. This is • Unicode: for Windows systems • non-Unicode: for all others This option is backwards compatible to older Runtimes.
		Unicode	Force Unicode mode.
		non-Unicode	Force non-Unicode mode.
Connection Pooling			
connectionPooling	Added in Builder 5.1.8.58 Runtime 5.1.82.0 This tagged value controls the connection pooling. If true, each connection is put into a pool after use. If an SQL adapter requires a connection, it is taken from the pool. If no connection is available, a new connection is being created and put into the pool after use. The time the connection is kept in the pool depends on the other pooling parameters.	true	Database connections are pooled.
		false	Database connections are not pooled.
maxConnectionAge	After a given connection age (in minutes) the connection will be closed and removed from the pool.	Connection age in minutes, default is **15 minutes**, **-1** means forever.	
maxConnectionIdleTime	Connections not used for the time specified (in minutes) will be closed and removed from the pool. This is useful for connections going through firewalls because such connections might be cut off after some time.	Values in minutes, default is **60**.	
maxConnectionReuse	This tagged value controls how often a connection can be re-used. After the connection has been re-used for **maxConnectionReuse**, it will be closed and not put back into the pool. This feature has been introduced because some databases had problems if the connection was re-used too often. Value **-1** means the connection will be re-used forever. In this case you should define reasonable values for **maxConnectionAge** or **maxConnectionIdleTime** (see above).	0	pooling is implicitly switched off.
		-1	connections are pooled forever
		a value	number of connections to be pooled, default is 1000.
	Note that the pooling is implicitly switched off, if **maxConnectionReuse** is set to 0.		
Qualifier			
schema	String that prefixes tables and stored procedures. For example, if schema is set to S1, all tables accessing the current DB are prefixed by "S1.". This works only if the tables are marked using the `TABLE::` keyword, e.g `TABLE::TEMPLOYEE` in SQL statements. If you do not prefix the table name by `TABLE::`, the tablename is used as it is.		

tableQualifier	String that prefixes tables. For example, if **tableQualifier** is set to TQ1, all tables accessing the current DB are prefixed by "TQ1", e.g. `TQ1TEMPLOYEE`. If schema and table qualifier are given, all tables will become: <schema>.<tableQualifier><tableName>. This works only if the tables are marked using the `TABLE::` keyword, e.g `TABLE::TEMPLOYEE` in SQL statements. If you do not prefix the table name by `TABLE::`, the tablename is used as it is.		