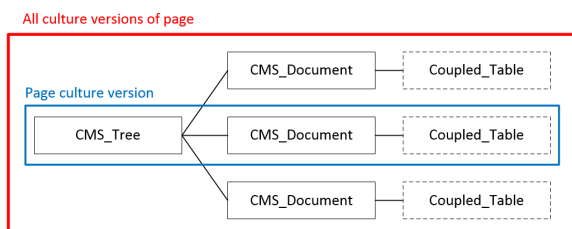


The Kentico database stores pages (nodes in the content tree of websites) in several interconnected tables:

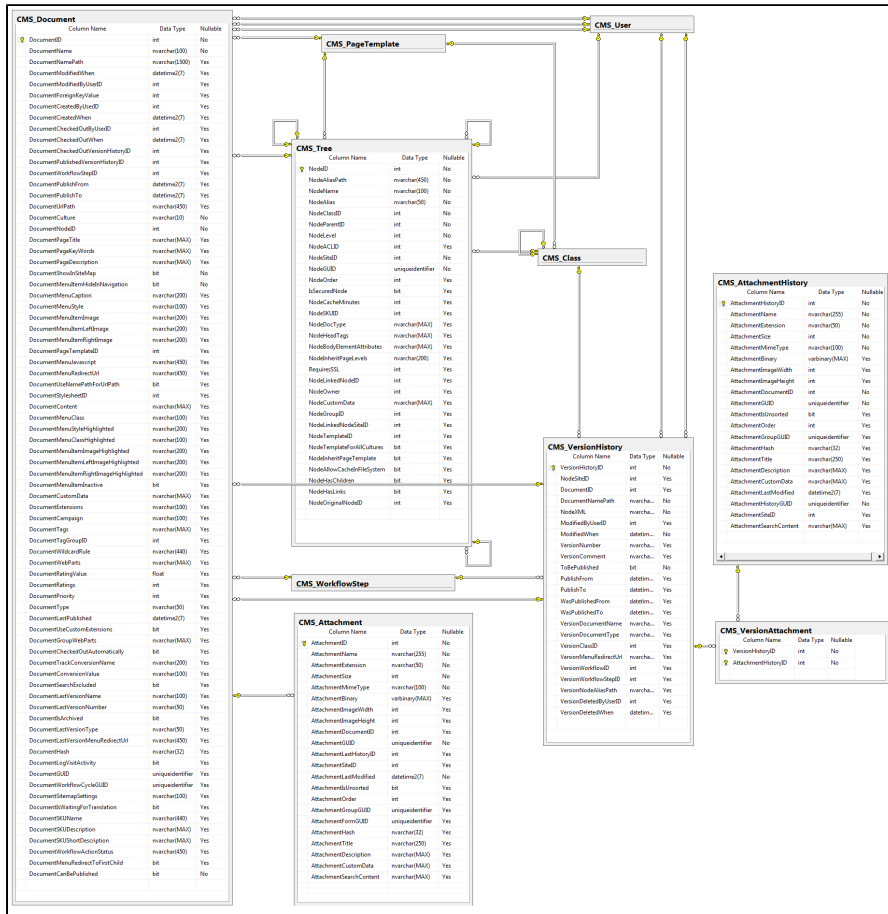
Database table	Description
CMS_Tree	<p>Stores:</p> <ul style="list-style-type: none"> <li>Data that determines the position of pages in the website content tree</li> <li>Basic page properties that are shared between all <a href="#">language versions</a> of the same page</li> </ul> <p>Contains <b>only one record for each page</b>, regardless of the number of language versions. The table doesn't store any <a href="#">versioned data</a>.</p>
CMS_Document	<p>Stores:</p> <ul style="list-style-type: none"> <li>Page properties that have unique values for each <a href="#">language version</a></li> <li>The content of <a href="#">editable regions</a></li> <li>The data of <a href="#">editor widgets</a></li> </ul> <p>May contain <b>multiple records per page</b>, one for each language version. Some of the page columns are <a href="#">versioned</a>.</p>
Coupled data tables (CONTENT_<doc type>)	<p>Stores the fields of individual <a href="#">page types</a>.</p> <p>For example, the <i>News</i> page type has the <b>CONTENT_News</b> coupled table with the columns defined for news pages, such as <i>NewsTitle</i>, <i>NewsSummary</i>, <i>NewsText</i>.</p> <ul style="list-style-type: none"> <li>Records in the coupled data tables are bound to pages through the <b>DocumentForeignKeyValue</b> column of the <b>CMS_Document</b> table.</li> <li>The page type of tree nodes is determined by the <b>NodeClassID</b> column in the <b>CMS_Tree</b> table.</li> <li>Container page types, such as <i>Folder</i>, do not have a coupled table.</li> <li>Every <a href="#">language version</a> of a page has its own record in the corresponding coupled data table.</li> <li>All columns in coupled data tables are <a href="#">versioned</a>.</li> </ul>



## Page attachments and versions

Database table	Description
----------------	-------------

CMS_Attachment	<p>Stores files uploaded to the database as <a href="#">page attachments</a>.</p> <p>Includes all types of page attachments:</p> <ul style="list-style-type: none"> <li>• Unsorted attachments added on the <i>Properties</i> -&gt; <i>Attachments</i> tab in the <i>Pages</i> application. Such attachments are linked to the related page by storing its identifier (<i>AttachmentDocumentID</i> column).</li> <li>• Attachments uploaded through specific page fields (of either the <i>File</i> or <i>Attachments</i> data type) on the <i>Form</i> or <i>Content</i> tab in the <i>Pages</i> application. Such fields are represented by columns in the given page type's coupled data table. <ul style="list-style-type: none"> <li>◦ Page type fields of the <i>File</i> data type store a GUID value, which identifies the corresponding attachment in the <i>CMS_Attachment</i> table.</li> <li>◦ For fields of the <i>Attachments</i> data type, the value in the database is always empty. Instead, the given records in the <i>CMS_Attachment</i> table store an identifier of the related page (<i>AttachmentDocumentID</i> column) and a GUID value that identifies the specific field in the page type's form definition (<i>AttachmentGroupGUID</i> column).</li> </ul> </li> </ul>
CMS_VersionHistory	<p>Stores <a href="#">page versions</a>, including older versions and versions that are being edited. When a version is published, the system updates the corresponding records in the <b>CMS_Tree</b> and <b>CMS_Document</b> tables using the data of the version record.</p>
CMS_AttachmentHistory	<p>Stores attachments of page versions. The main purpose of this table is to avoid redundancy for page versions that use the same attachments.</p> <p>When a page version is published, the system updates the records in the <b>CMS_Attachment</b> table using the data of the version record.</p>
CMS_VersionAttachment	<p>Stores relationships between page versions and their page attachments.</p>



## Products

**Products** represent merchandise or services in online stores. The system associates product objects (SKUs) with pages representing product types.

The primary product data is stored in the **COM\_SKU** database table.

- Pages representing products are bound to the related SKU through the **NodeSKUID** column of the **CMS\_Tree** table.
- Most product columns are **versioned** and covered by **workflow** (exceptions include the product image, as well as the *Track inventory* and *Available items* properties).
- If a product page is available in multiple **language versions** AND covered by workflow, the versioned product properties are **NOT shared between different language versions** of a single product. Any updates of the product **Price** (or other key properties) need to be made separately for each language version.

## Page relationships

Database table	Description
CMS_Relationship	Stores records representing relationships between two pages. Used for both: <ul style="list-style-type: none"> <li><b>Standard named relationships</b></li> <li>Ad-hoc relationships that represent re-use of a page's content within a <i>Pages</i> field of another page (see <b>Advanced content modeling</b>)</li> </ul>
CMS_RelationshipName	Stores the relationship names (types) defined in the system.



CMS\_RelationshipNameSite

Binding table that connects relationship names to websites. Each entry indicates that a relationship can be used on a given site.

