Ethan Dunham and Nathan Stauffer

Intro To Databases

Final Project

**Archaeology Database:**

Archaeology is the study of past peoples by looking at the remains they have left behind. Usually, this involves digging holes in the group, usually referred to as test units or shovel test pits. When artifacts aRe found, the area becomes a site. A site contains many artifacts and usually several test units.

The test pits are excavated by level, usually in 10cm increments. Of course, sometimes artifacts are on the surface. Since a site doesn’t need test units, but an artifact might be found on the surface, we need to add a default “surface” find. For the purposes of this project, the best place for that is as a default test unit that is added automatically when a new site is created.

We will be making a database that could be used for storing inventory of an archaeological curation facility. Each artifact must be separated by site, test unit, test unit level, FS number, artifact material type, date excavated and a short description. A test unit has to have a site. You cannot have a test unit without a site. Inside each test unit, there may be artifacts and artifacts must be made out of something (wood, metal, etc.). You can have an object made of multiple materials, like a hammer that is made of wood and metal.

**Database Outline:**

* Entity/table
  + attribute
    - description of attribute

**Contents of Database:**

* Sites:
  + Site Number:
    - A unique identifier for the site. Usually containing the county code, number of sites found that year, and the year. (16EBR2016 would signify the 16th site found in East Baton Rouge county Louisiana in 2016). This is the primary key.
  + Name:
    - A name for the site, usually unique, but does not have to be. This is what you would usually refer to the site in conversation (Abe Lincoln’s House).
  + State:
    - The 2 digit state code for the state that the site is located in (LA, CA…).
* Test Units:
  + Test Unit Number (tu\_number):
    - The number that designates that hole in that site. Usually, these are labeled starting with 1. Sometimes, a cardinal direction is added to the number to signify an extension to that test unit. This is part of the primary key.
  + Site ID:
    - The site number for the site this test unit is located at. Since number is only unique to the specific site, we are using a combination of Site ID and tu\_number as the primary key. This is part of the primary key.
* Artifacts:
  + Field Specimen Number(FS):
    - A unique number for each artifact or group of artifacts. (bags of the same type of glass found in the same general area usually have 1 FS number) This is the primary key for artifacts.
  + Site ID:
    - The site number that the artifact came from.
  + Test Unit Number:
    - The test unit number that the artifact came from.
  + Test Unit level:
    - Further divides each test unit by depth. Usually each test unit is divided into 10cm depth. Sometimes, a cardinal direction is added to them as well, but is not required. (tu:3 lvl:3E would be the 3rd test unit, 20-30cm down, on the east side.)
  + Date Excavated:
    - The month, day, and year that the artifact was removed from the ground.
  + Description:
    - A short string that describes the artifact, usually unique.
* Material Type:
  + ID:
    - Integer ID for that material type. This is the primary key.
  + Name:
    - The name of that material (wood, bone, stone, metal).

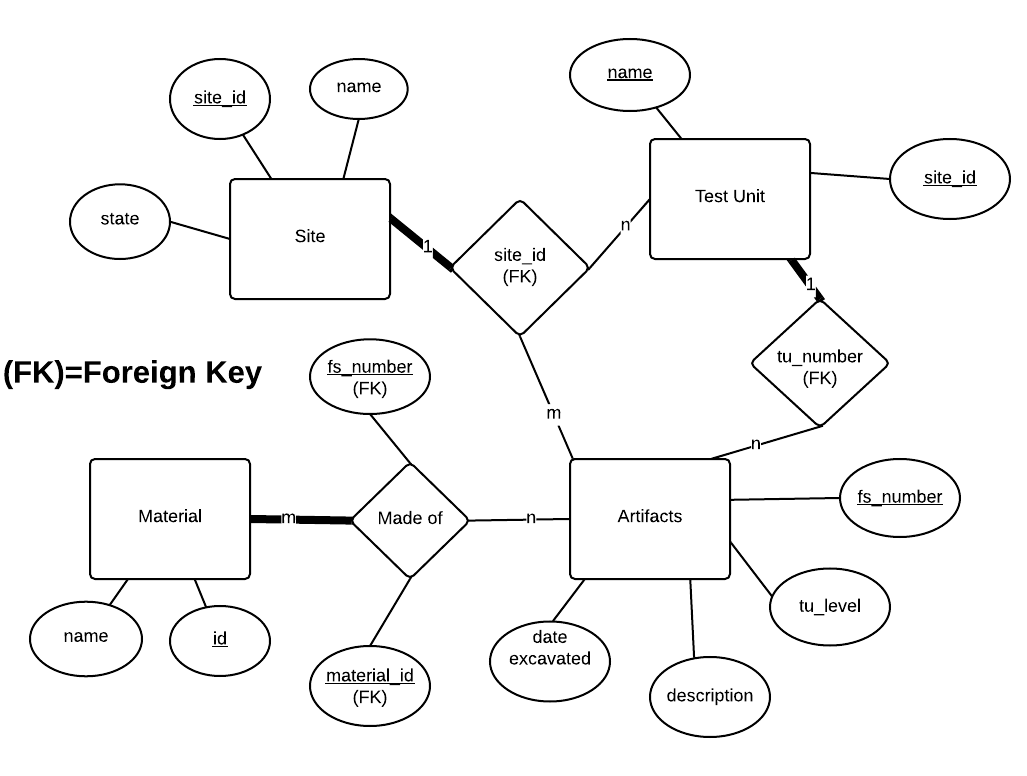
**Extra table for Many to Many Relationship:**

* Artifact Material Table:
  + fs\_number:
    - Artifact FS number. This is part of the primary key.
  + Material Type ID:
    - The ID of the material type that the artifact is made of. This is part of the primary key.

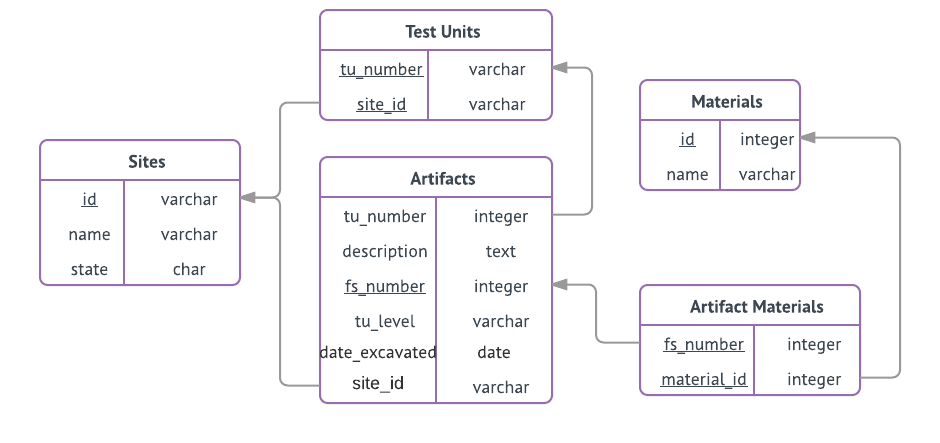
**Relationships:**

* One to Many
  + Sites can contain many test units, but each test unit only has one site.
  + Sites can contain many artifacts, but each artifact only has one site.
  + Test units can contain many artifacts, but artifacts only have 1 test unit.
* Many to Many
  + Artifacts can be made of many different materials and a material can describe many artifacts.

**ER Diagram:**

****

**Schema:**



**Data Definition Queries:**

SET FOREIGN\_KEY\_CHECKS = 0;

DROP TABLE IF EXISTS `artifacts`;

DROP TABLE IF EXISTS `sites`;

DROP TABLE IF EXISTS `test\_units`;

DROP TABLE IF EXISTS `artifact\_materials`;

DROP TABLE IF EXISTS `materials`;

CREATE TABLE `sites`(

`id` varchar(255) NOT NULL,

`name` varchar(255),

`state` char(2),

PRIMARY KEY (id)

)ENGINE=innodb;

CREATE TABLE `test\_units`(

`tu\_number` varchar(255) NOT NULL,

`site\_id` varchar(255) NOT NULL,

PRIMARY KEY (tu\_number, site\_id),

FOREIGN KEY(site\_id) REFERENCES sites (id) ON DELETE CASCADE ON UPDATE CASCADE

)ENGINE=innodb;

CREATE TABLE `artifacts`(

`fs\_number` int(11) NOT NULL,

`site\_id` varchar(255) NOT NULL,

`description` text,

`tu\_number` varchar(255) NOT NULL,

`tu\_level` varchar(255),

`date\_excavated` date,

PRIMARY KEY (fs\_number),

FOREIGN KEY(tu\_number) REFERENCES test\_units (tu\_number) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(site\_id) REFERENCES sites (id) ON DELETE CASCADE ON UPDATE CASCADE

)ENGINE=innodb;

CREATE TABLE `materials`(

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255),

PRIMARY KEY (id)

)ENGINE=innodb;

CREATE TABLE `artifact\_materials`(

`fs\_number` int(11) NOT NULL,

`material\_id` int(11) NOT NULL,

PRIMARY KEY (fs\_number, material\_id),

FOREIGN KEY(fs\_number) REFERENCES artifacts (fs\_number) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(material\_id) REFERENCES materials (id) ON DELETE CASCADE ON UPDATE CASCADE

)ENGINE=innodb;

SET FOREIGN\_KEY\_CHECKS = 1;

**Data Manipulation Queries:**

* **To populate sites table on homepage:** SELECT id, name, state FROM sites
* **To populate test units table on homepage:** SELECT tu\_number, site\_id FROM test\_units
* **To populate artifacts table on homepage:** SELECT fs\_number, tu\_number, tu\_level, site\_id, date\_excavated, description FROM artifacts
* **To populate artifact materials table on homepage:** SELECT am.fs\_number, m.id FROM artifact\_materials am INNER JOIN materials m ON am.material\_id = m.id
* SELECT tu\_number FROM test\_units WHERE site\_id='[$site]'
* **To add a new artifact:** INSERT INTO artifacts (fs\_number, description, tu\_number, site\_id, tu\_level, date\_excavated) VALUES ('[$fs]', '[$description]', '[$tu]', '[$site]', '[$lvl]', '[$date]')
* **To add a new artifact/material relationship:** INSERT INTO artifact\_materials (fs\_number, material\_id) VALUES ('[$fs]', '[$mats]')
* **To add a new artifact/material relationship with multiple materials:** INSERT INTO artifact\_materials (fs\_number, material\_id) VALUES ('[$fs]', '[$mat]')
* **To add a new material:** INSERT INTO materials (name) VALUES ('[$material]')
* **To add a new site:** INSERT INTO sites (id, name, state) VALUES ('[siteID]', '[$siteName]', '[$state]')
* **To add a new test unit level:** INSERT INTO test\_units (tu\_number, site\_id) VALUES ('Surface', '[$siteID]')
* **To add a new test unit:** INSERT INTO test\_units (tu\_number, site\_id) VALUES ('[$tu]','[$site]')
* **To edit site info:** UPDATE sites SET id='[site]', name='[$name]', state='[$state]' where id='[$oldSite]'
* **To populate artifact dropdown:** SELECT fs\_number, description FROM artifacts
* **To populate site name dropdown:** SELECT name FROM sites WHERE id='[$site]'
* **To populate site state dropdown:** SELECT state FROM sites WHERE id='[$site]'
* **To filter by material type:** SELECT a.fs\_number, a.description, s.name, a.tu\_number, a.tu\_level, a.date\_excavated FROM artifacts a INNER JOIN sites s ON a.site\_id = s.id INNER JOIN artifact\_materials am ON a.fs\_number = am.fs\_number INNER JOIN materials m ON am.material\_id = m.id WHERE m.id=[?]
* **To remove a material:** DELETE FROM materials WHERE id=[$name]
* **To populate sites dropdown:** SELECT id, name FROM sites
* **To populate materials dropdown:** SELECT id, name FROM materials
* **To filter by site:** SELECT s.name, a.fs\_number, a.description, a.tu\_number, a.tu\_level, a.date\_excavated FROM artifacts a INNER JOIN sites s ON a.site\_id = s.id WHERE s.id=[?]

**Website URL:**

<https://web.engr.oregonstate.edu/~dunhamet/databasefinal/main_page.php>