Augur Table	**Description**	**Type of Data**
chaoss_metric_status	Status information on Augur Implementation of CHAOSS Defined Metrics	Status table for Augur implementation of CHAOSS Metrics
commit_comment_ref	Bridge table between commit comments and shared "messages" table.	Messages bridge tables
commits	A detailed record of all commits in a repository.	Core
contributor_affiliations	Organizational affiliations for repository contributors, with begin dates and end dates. This requires some degree of maintenance from the organization overseeing the repository.	Individual Aliasing
contributors	Each person who contributes an issue, pull request (or similar synonym from a non-GitHub system), commit or comment on any of the above is recorded as a repository contributor. The types of contributions are derived through the API most conveniently.	Core
contributors_aliases	Many .git repository contributors have more than one email address. When we find an email address shared across data sources, we are increasingly automating the aliasing of the same indvidual to multiple emails.	Individual Aliasing
dm_repo_annual	Summary tables for commit data. Since this data can become quite large, we summarize it for presentation.	Summary
dm_repo_group_annual		Summary
dm_repo_group_monthly		Summary
dm_repo_group_weekly		Summary
dm_repo_monthly		Summary
dm_repo_weekly		Summary
issues	From an issue tracker.	Core
issue_assignees	People assigned to work on the issue in the issue tracker. The state of this list is routinely updated.	Supporting Detail
issue_events	An enumeration of all events on an issue, including changes in status, assignees and lables	Supporting Detail
issue_labels	Labels assigned to each each issue in a repository.	Supporting Detail

Augur Table	Description	Type of Data
issue_message_ref	Bridge table between issue comments and shared "messages" table.	Messages bridge tables
libraries	Future: A implementation libraries.io like dependency features. Currently we access this information with some amount of non-automated labor using Libraries.io data dumps. We have developed strategies for cleaning this data.	Core
library_dependencies		Supporting Detail
library_version		Supporting Detail
message	A table where all repository messages, including comments on commits, pull requests and issues, are stored together. We chose this article to enable ocmputational linquisitic analysis of all written communication in a repository ecosystem more accessible.	Core
platform	Where did the data come from? GitHub? Bugzilla? Each platform as a row in this table.	Metadata reference
pull_requests	Referred to generally as "reviews" in the CHAOSS metrics, many popular .git based platforms support forking and pulling.	Core
pull_request_assignees	An up to date listing of individulas assigned to pull requests.	Supporting Detail
pull_request_events	An enumeration of all events on a pull reuest, including changes in status, assignees and labels.	Supporting Detail
pull_request_labels	Labels assigned to each each pull request in a repository.	Supporting Detail
pull_request_message_ref	Bridge table between pull request comments and shared "messages" table.	Messages bridge tables
pull_request_meta		Supporting Detail
pull_request_repo	The repository the pull request originates from.	Supporting Detail
pull_request_reviewers	Contributors assigned to the role of pull request reviewer.	Supporting Detail
pull_request_teams	Teams assigned to pull requests.	Supporting Detail
repo	"The Git Repository." Star of the show.	Core
repo_groups	Groups of repositories, often a GitHub, GitLab or other organizational configuration, that open source program offices or community managers consider to be "parts of a larger project".	Summary
repo_badging	A table that implements the Linux Foundation's Core Infrastructure Initiative's ~300 questions used to determine whether or not a project is badged. Current around 3,000 total repositories are part of this process.	Supporting Detail

Augur Table	Description	Type of Data
repo_insights	Raw data used by accessing Augur endpoints to identify hashed combinations of metrics at a point in time, at the repo level.	Machine learning based anomaly detection, used to power push notifications
repo_group_insights	Raw data used by accessing Augur endpoints to identify hashed combinations of metrics at a point in time, at the repo_group level.	
repo_insights_records	Stored instances of the most recent anomalies for a metric in a repository. Today the normal range is determined using the prior 12 months, and anomalies are reported during the immediately preceding 90 days. Both will be parameterizable in future releases, and are used to power our "Slack Bot"	
repo_groups_list_serve	An enumeration of list-serves related to a repository.	
repo_info	A collection of summary data, collected each day for each repository, including the number of forks, watches, issues, open issues, pull requests, and other information used for some summary statistics and also validation of Augur's collection of the raw data.	Supporting Detail
repo_labor	Future: An automated implementation the Cocomo code counting and complexity measurement algorithm. Currently we provide this information with some amount of non-automated labor.	Supporting Detail
repo_meta	A convenience table for creating name-value pairs of metrics desired in customized Augur implementations	Supporting Detail
repo_sbom_scans	Software bill of materials scans provided by Augur-SBOM (formerly DoSOCS)	Supporting Detail
repo_test_coverage	Future: Statistics on code coverage and method coverage of testing, as defined in CHAOSS Metrics	Supporting Detail