Object Drive 1.0
Metadata and File System Structure

Last Updated on September 13, 2019

Object Drive leverages a relational database (RDS) for storage of metadata associated with objects in the system. For file streams, content may be stored on disk optionally encrypted at rest, and this can map to an S3 bucket.

This document is intended to provide guidance on the structure of data within the database, and how it correlates to files stored on the file system. This is useful information for accessing data stored by Object Drive without going through the API, as may be the case for either bulk data migrations, data ingestion, or performing analytics on an unencrypted file store.

Metadata Database

The current list of tables used by Object Drive is listed below

Table	Description
a_object	Archive table for all revisions to an object, treated as immutable
a_object_permission	Archive table for all revisions to an object permission, treated as immutable
a_object_type	Archive table for all revisions to an object type, treated as immutable
a_property	Archive table for all revisions to a property, treated as immutable
a_user	Archive table for all revisions to a user, treated as immutable
acm2	Encapsulates unique ACMs in use
acmgrantee	Provides Correlation between a resource string to individual fields of an acm share and the flattened value from AAC
acmkey2	Lookup values for named ACM field keys
acmpart2	Provides correlation from an ACM to a key and value referenced by

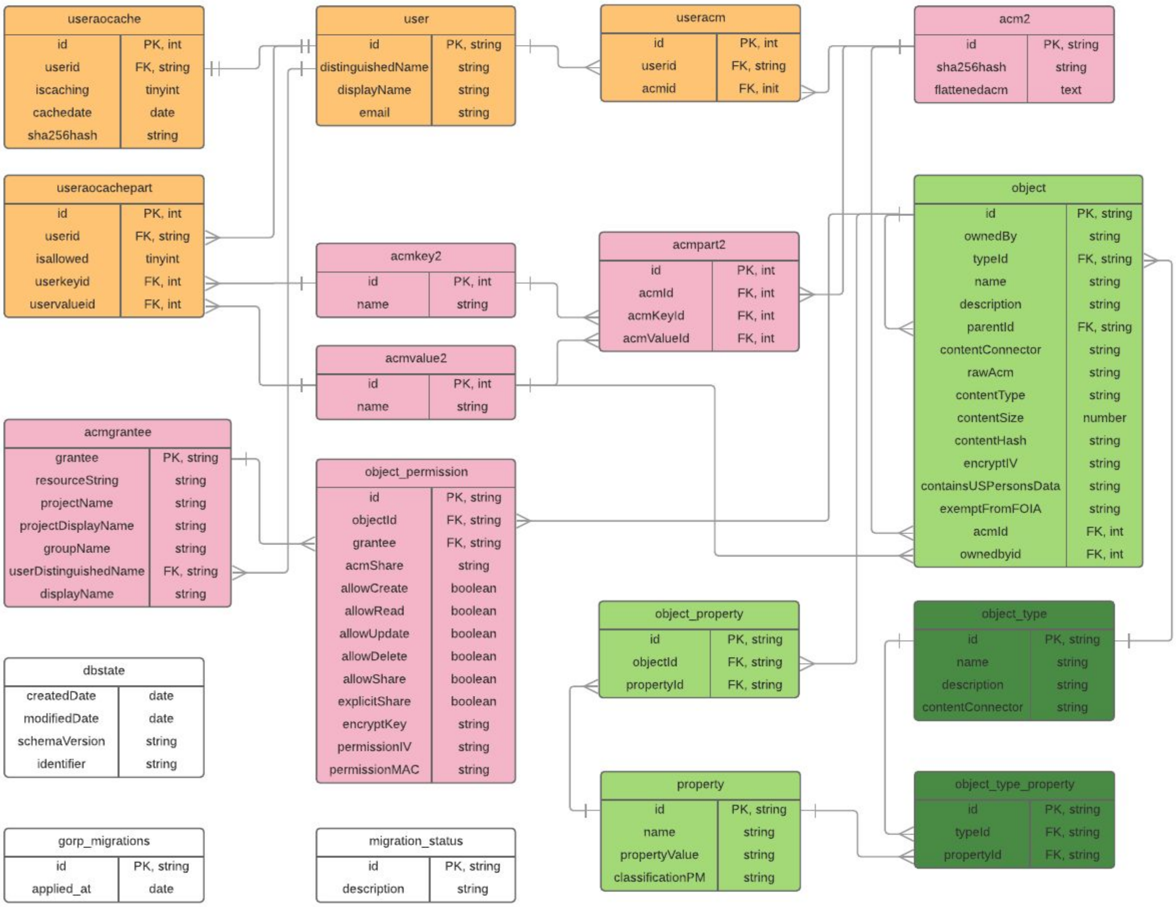
	IDs
acmvalue2	Lookup values for ACM field values
dbstate	Contains a single record denoting metadata about the schema and instance identification
gorp_migrations	Provides list of all migrations applied to the database schema and when. Used by the odrive-database migration tool
migration_status	Provides a place for long running migrations to write incremental steps that may be reported by the odrive-database migration tool
object	Current state of objects
object_permission	Current permissions for objects
object_property	Associates objects to properties
object_type	Current state of object types
object_type_property	Associates object types to properties
property	Current state of properties, whether for an object or a type
user	Metadata about a user primarily for capturing distinguished name
useracm	Fast lookup of associations between users and known acms
useraocache	Local cached information about a user's authorization object policy
useraocachepart	Association of rules from AAC provided snippets to a user by correlating to the same named ACM field keys and values.

Archive tables begin with the prefix "a_", and have a data structure exactly the same as their current status table counterparts, with the addition of an surrogate primary key identifier for uniqueness amongst versions, and the lack of foreign key constraints and triggers.

Archive tables are populated through the execution of insert and delete triggers on their current status table counterparts. This allows for automatic revision creation for every update in a consistent way, and automated hash to generate change tokens.

Entity Relationship Diagram

Depicted below is a simplified entity relationship diagram (ERD) of all tables except for archive tables, and only depicting non-common fields to reduce noise.



useraocache	
id	PK, int
userid	FK, string
iscaching	tinyint
cachedate	date
sha256hash	string

user	
id	PK, string
distinguishedName	string
displayName	string
email	string

useracm	
id	PK, int
userid	FK, string
acmid	FK, int

acm2	
id	PK, string
sha256hash	string
flattenedacm	text

useraocachepart	
id	PK, int
userid	FK, string
isallowed	tinyint
userkeyid	FK, int
uservalueid	FK, int

acmkey2	
id	PK, int
name	string

acmpart2	
id	PK, int
acmId	FK, int
acmKeyId	FK, int
acmValueId	FK, int

acmvalue2	
id	PK, int
name	string

object	
id	PK, string
ownedBy	string
typeId	FK, string
name	string
description	string
parentId	FK, string
contentConnector	string
rawAcm	string
contentType	string
contentSize	number
contentHash	string
encryptIV	string
containsUSPersonsData	string
exemptFromFOIA	string
acmId	FK, int
ownedbyid	FK, int

acmgrantee	
grantee	PK, string
resourceString	string
projectName	string
projectDisplayName	string
groupName	string
userDistinguishedName	FK, string
displayName	string

object_permission	
id	PK, string
objectId	FK, string
grantee	FK, string
acmShare	string
allowCreate	boolean
allowRead	boolean
allowUpdate	boolean
allowDelete	boolean
allowShare	boolean
explicitShare	boolean
encryptKey	string
permissionIV	string
permissionMAC	string

object_property	
id	PK, string
objectId	FK, string
propertyId	FK, string

object_type	
id	PK, string
name	string
description	string
contentConnector	string

dbstate	
createdDate	date
modifiedDate	date
schemaVersion	string
identifier	string

property	
id	PK, string
name	string
propertyValue	string
classificationPM	string

object_type_property	
id	PK, string
typeId	FK, string
propertyId	FK, string

gorp_migrations	
id	PK, string
applied_at	date

migration_status	
id	PK, string
description	string

The common fields that are included in several tables are

- createdDate - timestamp when record created
- createdBy - user that created the record, foreign key to user.distinguishedName
- modifiedDate - timestamp when record last modified
- modifiedBy - user that last modified the record, foreign key to user.distinguishedName
- isDeleted - bit flag indicating if record is deleted
- deletedDate - timestamp when record was marked deleted, null if not deleted
- deletedBy - user that deleted the record, foreign key to user.distinguishedName
- changeCount - total changes, 0 based for record created and unchanged
- changeToken - md5 hash of record id, modified date, and change count

Core Tables

Object

Stores the current state of objects. Revision information is stored in the a_object table.

column	data type	description
id	byte(16)	Primary Key Holds the GUID acting as the primary key for an object for identification. String representation throughout the application and API is a 32 character long hexadecimal representation.
createdDate	timestamp(6)	A timestamp in UTC for when this object was created.
createdBy	varchar(255)	Stores the normalized distinguished name for the user that created this object
modifiedDate	timestamp(6)	A timestamp in UTC for when this object was most recently modified. When an object is first created, this value will reflect the same value as that stored in createdDate.
modifiedBy	varchar(255)	Stores the normalized distinguished name for the user that most recently modified this object. When an object is first created, this value will reflect the same as that stored for createdBy.
isDeleted	tinyint(1)	A bit flag indicating whether this object has been marked as deleted.
deletedDate	timestamp(6)	A timestamp in UTC for when this object was deleted. A value of null indicates that it is not deleted.
deletedBy	varchar(255)	The normalized distinguished name for the user that marked the object as deleted. If the current state of the object is not deleted, then it's value will be null.
isAncestorDeleted	tinyint(1)	A bit flag indicating whether this object has been marked as deleted implicitly due to an ancestor being explicitly deleted. This relates to the delete and undelete of trees of data if you delete a folder.
isExpunged	tinyint(1)	A bit flag indicating whether this object has been marked as deleted forever. Objects in this state cannot be restored with the API, but still exist in the database to permit them being restored by an administrator, or for facilitating purges.
expungeDate	timestamp(6)	A timestamp in UTC for when this object was expunged.
expungedBy	varchar(255)	The normalized distinguished name for the user that marked

		the object as expunged. If the current state of the object is not expunged, then its value will be nul.
changeCount	int(11)	Denotes the revision number for an object. This value is 0 based, meaning that an object that is created will have a changeCount of 0, with each revision incrementing the count
changeToken	varchar(60)	Holds the expected value to permit an update to the record. The change token is calculated when records are created and updated, and is a MD5 Hash of the concatenation string representations of the record id, its change count, and modified date.
ownedBy	varchar(255)	Contains the resource string of the owner of the object. An owner is one entity, whether it is a single user, or a group. Resource strings are discussed in later sections.
typeId	binary(16)	Refers to the GUID of a related object_type record for the object type identification.
name	varchar(255)	The given file or folder name for an object.
description	varchar(255)	An abstract of the object or its contents.
parentId	binary(16)	When populated, it indicates that this object is the child of the object identified by the GUID value given. In this way, objects may be given a hierarchy.
contentConnector	varchar(2000)	This field was originally intended to store permanent storage URI and access information. It currently is used for a single access pattern for files as stored in local cache, and in S3 backend. More details about the usage of this value in local cache and S3 are discussed in later sections.
rawAcm	text	The full ACM as provided, validated and populated in JSON format.
contentType	varchar(255)	The mime-type of the file's content stream
contentSize	bigint(20)	The length of a content stream in bytes
contentHash	binary(32)	A SHA256 hash of the plaintext of the content stream in a hexadecimal encoded string.
encryptIV	binary(16)	The initialization vector for encrypting the content stream for this object.
containsUSPersonsData	varchar(255)	Indicates if this object contains US persons data.
exemptFromFOIA	varchar(255)	Indicates if this object is exempt from freedom of information act requests.
acmid	int(10) unsigned	Numeric identifier of the immutable ACM association, foreign

		key association to acm2 table.
ownedbyid	int(10) unsigned	Numeric identifier of the owner of the record, determined by the correlation of the grantee from acmgrantee table to value in acmvalue2, using the acmvalue2 id for foreign key association.

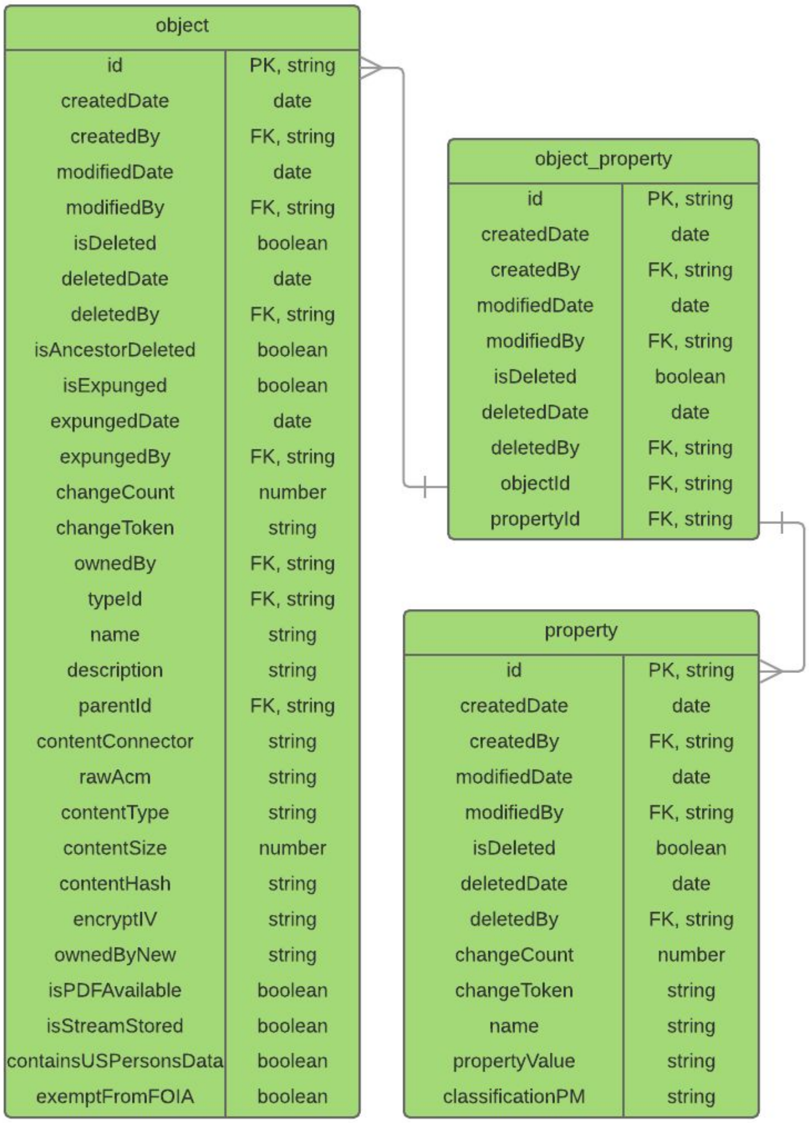
Property

Stores the current state of properties which are in turn associated with an object.

column	data type	description
id	byte(16)	Primary Key Holds the GUID acting as the primary key for this property for identification. String representation throughout the application and API is a 32 character long hexadecimal representation.
createdDate	timestamp(6)	A timestamp in UTC for when this property was created.
createdBy	varchar(255)	Stores the normalized distinguished name for the user that created this property
modifiedDate	timestamp(6)	A timestamp in UTC for when this property was most recently modified. When a property is first created, this value will reflect the same value as that stored in createdDate.
modifiedBy	varchar(255)	Stores the normalized distinguished name for the user that most recently modified this property. When a property is first created, this value will reflect the same as that stored for createdBy.
isDeleted	tinyint(1)	A bit flag indicating whether this property has been marked as deleted.
deletedDate	timestamp(6)	A timestamp in UTC for when this property was deleted. A value of null indicates that it is not deleted.
deletedBy	varchar(255)	The normalized distinguished name for the user that marked the property as deleted. If the current state of the property is not deleted, then it's value will be null.
changeCount	int(11)	Denotes the revision number for a property. This value is 0 based, meaning that a property that is created will have a changeCount of 0, with each revision incrementing the count
changeToken	varchar(60)	Holds the expected value to permit an update to the record. The change token is calculated when records are created and updated, and is a MD5 Hash of the concatenation string representations of the record id, its change count, and

		modified date.
name	varchar(255)	The name of the property
propertyValue	varchar(20000)	The value of the property
classificationPM	varchar(200)	A dedicated field intended for storage of the portion mark for the value of a property. This field is not required, and no checking or validation is performed on this field either individually, or for rollup rules enforcement on the object rawacm.

An object has associations to its properties as modeled below



object	
id	PK, string
createdDate	date
createdBy	FK, string
modifiedDate	date
modifiedBy	FK, string
isDeleted	boolean
deletedDate	date
deletedBy	FK, string
isAncestorDeleted	boolean
isExpunged	boolean
expungedDate	date
expungedBy	FK, string
changeCount	number
changeToken	string
ownedBy	FK, string
typeId	FK, string
name	string
description	string
parentId	FK, string
contentConnector	string
rawAcm	string
contentType	string
contentSize	number
contentHash	string
encryptIV	string
ownedByNew	string
isPDFAvailable	boolean
isStreamStored	boolean
containsUSPersonsData	boolean
exemptFromFOIA	boolean

object_property	
id	PK, string
createdDate	date
createdBy	FK, string
modifiedDate	date
modifiedBy	FK, string
isDeleted	boolean
deletedDate	date
deletedBy	FK, string
objectId	FK, string
propertyId	FK, string

property	
id	PK, string
createdDate	date
createdBy	FK, string
modifiedDate	date
modifiedBy	FK, string
isDeleted	boolean
deletedDate	date
deletedBy	FK, string
changeCount	number
changeToken	string
name	string
propertyValue	string
classificationPM	string

Object_Permission

Stores the current permissions for an object.

column	data type	description
id	byte(16)	Primary Key Holds the GUID acting as the primary key for this permission for identification. String representation throughout the application and API is a 32 character long hexadecimal

		representation.
createdDate	timestamp(6)	A timestamp in UTC for when this permission was created.
createdBy	varchar(255)	Stores the normalized distinguished name for the user that created this permission
modifiedDate	timestamp(6)	A timestamp in UTC for when this permission was most recently modified. When a permission is first created, this value will reflect the same value as that stored in createdDate.
modifiedBy	varchar(255)	Stores the normalized distinguished name for the user that most recently modified this permission. When a permission is first created, this value will reflect the same as that stored for createdBy.
isDeleted	tinyint(1)	A bit flag indicating whether this permission has been marked as deleted.
deletedDate	timestamp(6)	A timestamp in UTC for when this permission was deleted. A value of null indicates that it is not deleted.
deletedBy	varchar(255)	The normalized distinguished name for the user that marked the permission as deleted. If the current state of the permission is not deleted, then it's value will be null.
changeCount	int(11)	Denotes the revision number for a permission. This value is 0 based, meaning that a permission that is created will have a changeCount of 0, with each revision incrementing the count
changeToken	varchar(60)	Holds the expected value to permit an update to the record. The change token is calculated when records are created and updated, and is a MD5 Hash of the concatenation string representations of the record id, its change count, and modified date.
objectId	binary(16)	Identifier for object to which this permission applies. Foreign key to object table.
grantee	varchar(255)	The flattened representation of a user or group. Foreign key to acmgrantee table.
acmShare	text	A JSON representation of the acm share block for this permission irrespective of the allowRead condition.
allowCreate	tinyint(1)	A bit flag indicating if the grantee will be given create
allowRead	tinyint(1)	A bit flag indicating if the grantee will be given read
allowUpdate	tinyint(1)	A bit flag indicating if the grantee will be given update
allowDelete	tinyint(1)	A bit flag indicating if the grantee will be given delete

allowShare	tinyint(1)	A bit flag indicating if the grantee will be given share
explicitShare	tinyint(1)	A bit flag indicating if this permission was explicitly defined during a create or update, or if it was implicitly given through the creation of an object that inherited permissions of its parent.
encryptKey	binary(32)	A unique encryption key for encrypting/decrypting the contnet stream of the object at rest for this grantee and revision.
permissionIV	binary(32)	A fresh random bitstring used for encrypting the key and implicitly the signature of the encrypt
permissionMAC	binary(32)	The permission message authentication code (MAC) allows for authenticating that the service wrote this permission.
createdbyid	int(10) unsigned	Numeric identifier of the creator of the record for faster joins to support filtering of permissions that were created by a user, and hence deriving objects shared by a user
granteeid	int(10) unsigned	Numeric identifier of the grantee of the record for faster joins to support filtering of permissions that apply to a user or group

Resource String Format

Various bits of the Object Drive logic and API calls make reference to a resource string format. This is a normalized format that allows for mapping users or groups as declared in the resource string back to the structure required for the share portion of an ACM

A resource string takes the following form:

    {resourceType}/{serialized-representation}/{optional-display-name}

There are two resourceType values: user and group

Examples for Users

- user/{distinguishedName}/{displayName}
- user/cn=test tester10,ou=people,ou=dae,ou=chimera,o=u.s. government,c=us/test tester10

Examples for groups

- group/{projectName}/{groupName}
- group/{projectName}/{projectDisplayName}/{groupName}/{displayName}

- group/dctc/odrive_g1
- group/dctc/DCTC/ODrive_G1/DCTC ODrive_G1
- group/-Everyone

How Content Connector is used for Local Cache

When a file is uploaded to Object Drive, there are different phases it transitions through in the local cache. On disk, files are stored in the location represented as

> {OD_CACHE_ROOT}/{OD_CACHE_PARTITION}/{dbID}/{filename}.{cachestate}

where
- OD_CACHE_ROOT, which defaults to "." is the absolute or relative path to set the root of the local cache settings
- OD_CACHE_PARTITION, which defaults to "cache" is an optional partition facilitating segregation of data amongst different instances of Object Drive.
- dbID is the database generated unique identifier for the metadata storage
- filename is the value of the contentConnector stored in the object table
- cachstate is one of the following states
    - uploading - When the file is being uploaded and written to disk in chunks
    - uploaded - When uploading has been completed, but not yet stored in permanent (s3) storage
    - caching - When a file is being retrieved from permanent (s3) storage, but not yet completely retreived.
    - cached - When the file has been fully retrieved from permanent (s3) storage to local cache, or when the file has finished being sent to permanent storage.

A file being uploaded goes through these states
1. When a file is uploaded to object drive, it will be assigned a newly randomized contentConnector value as the filename reference. Within local cache it will create and append to the file with the extension of .uploading
2. Once completed uploading, this file is renamed to .uploaded
3. A background process will then be responsible for getting that file stored in permanent storage.
4. If it is in permanent storage, its extension will then change to .cached

A file being downloaded may go through these states
1. The service checks to see if it has the file in local cache with either a .cached or .uploaded extension. If it does, it will serve that file
2. If it doesn't, then it will check peer nodes within the cluster or the same file. If any of those have it locally, they will serve that file

3. If neither the node or its peers have the file, then the service will pull the file from s3, with an initial extension of .caching.
4. Once completely retrieved the extension will be renamed to .cached

How Content Connector is used for S3 Storage

When a file is stored in S3, it will have the filename as defined by the ContentConnector metadata field, but without an extension. The effective URI is defined thusly

> s3://{OD_AWS_S3_BUCKET}/{OD_CACHE_PARTITION}/{dbID}/{filename}

where

- OD_AWS_S3_BUCKET is the environment variable defining the name of an S3 bucket to read and write to.
- OD_CACHE_PARTITION, which defaults to "cache" is an optional partition facilitating segregation of data amongst different instances of Object Drive.
- dbID is the database generated unique identifier for the metadata storage
- filename is the value of the contentConnector stored in the object table

Sample Database Queries

File metadata from an S3 filename

The use case for this query is primarily for those instances leveraging the ability to run with an unencrypted data store, to perform analytics on files stored in the S3 bucket. Starting from the key name, this query will bring back important information for processing and associating derived data.

Sample key:

lucasmoten/3a8d7ed21690-28ceec04/cea4a66ebea84270c0cffc30fb5994c4afd4ed6f41e60612691e930725b1d7ec

Database Query:

```
SELECT
       hex(id),
       name,
        contentType,
        contentSize
FROM
        object
WHERE
        ${thekey} like concat('%',contentConnector);
```

Replace ${thekey} with an actual key. If using the sample key above, the query would be

```
select hex(id), name, contentType, contentSize from object where
'lucasmoten/3a8d7ed21690-28ceec04/cea4a66ebea84270c0cffc30fb5994c4afd4ed6f41e60612691e930725b1d7ec' like
concat('%',contentConnector);
```

Sample Result

```
+----------------------------------+---------------+-------------+-------------+
| hex(id)                          | name          | contentType | contentSize |
+----------------------------------+---------------+-------------+-------------+
| 11E930A64847D8CDAA4E0242AC120005 | spacexvid.mp4 | video/mp4   |           0 |
+----------------------------------+---------------+-------------+-------------+
```

Get list of Content Connectors for a file type

The use case for this query is primarily for those instances leveraging the ability to run with an unencrypted data store, to perform analytics on files stored in the S3 bucket, but starting with a list of known files of a given type. For example, if you only want to process files that are images, then you start by interrogating the database to get a list of keys

Required Inputs

- OD_AWS_S3_BUCKET - This environment variable reflects the S3 bucket that contains the data
- OD_CACHE_PARTITION - This value partitions data at the root of the S3 bucket. If this value is not known, it may be the default `cache`

Database Query

```
SELECT
     CONCAT('${OD_CACHE_PARTITION}','/',dbstate.identifier,'/',object.contentconnector) s3key,
     hex(object.id) objectid,
     object.name name,
     object.contentSize size
FROM
     object, dbstate
WHERE
     object.contentType LIKE 'image/%';
```

Replace ${OD_CACHE_PARTITION} with the actual environment value. If using the sample value `cache` above, the query would be

```
SELECT CONCAT('cache','/',dbstate.identifier,'/',object.contentconnector) s3key,
hex(object.id) objectid, object.name name, object.contentSize size FROM object, dbstate WHERE
object.contentType LIKE 'image/%';
```

Sample Result

```
+----------------------------------------------------------------------------------------------+----------------------------------+----------------------------------+--------+
| s3key                                                                                        | objectid                         | name                             | size   |
+----------------------------------------------------------------------------------------------+----------------------------------+----------------------------------+--------+
| cache/3a8d7ed21690-28ceec04/399a5b42ed97411ad4b22e43ca51e5dd6463821fa33fbc0e0ec2ff85131923a1 | 11E930A6480528B1AA4E0242AC120005 | animated.gif                     | 313760 |
| cache/3a8d7ed21690-28ceec04/88db5a093cbd732b52fd3ec84d5bbd1526f0d47fc3f5a727e54c0510cd27d6a9 | 11E930A648308278AA4E0242AC120005 | diagram-gm-data.png              |  71329 |
| cache/3a8d7ed21690-28ceec04/cda940c620705d3b07622a2836ee81a1b834b55101a665cf7ff010fd00f7e3a8 | 11E930A64FA93584AA4E0242AC120005 | jira-DIMEODS-1183-1550182647.png |      0 |
+----------------------------------------------------------------------------------------------+----------------------------------+----------------------------------+--------+
```