

Data Dictionary

General Notes: The size of the field will only be specified when it is not equal to the maximum allowed by the data type. Format and Examples will also be omitted from int types because they are self-explanatory. For all string fields varchar was chosen unless the text was pulled from an external source, in which case text type was chosen for more flexibility. For the varchar fields, names of things were limited to a size of 500 to save space because names tend to be short while allowing for the possibility of extreme cases, and descriptions were given the maximum size of 8000 because these can be longer string data but since they will be written by me and not pulled from elsewhere the 8000-byte limit is acceptable. Boolean was used to indicate a particular category or feature of an entity when there were only two choices and no possibility of adding more, otherwise varchar was used to indicate categories and the possible categories were listed in the description.

TABLE: Series

- Field: id
 - *Data Type:* Integer, autonumber
 - *Constraint:* Primary Key, Not Null
 - *Description:* Unique numeric identifier

Note: In the interest of saving page space, autonum primary key id fields will be omitted from now on.

- Field: title
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Title Case”
 - *Constraint:* Not Null
 - *Description:* The title of the comic book series. Can be found in the reading order sites listed next to the issue number. Do not specify volume number.
 - *Example:* “Action Comics”
- Field: publication_year
 - *Data Type:* date
 - *Data Format:* YYYY. Standard: [W3CDTF](#). Standard was chosen because it is a widely used international date standard.
 - *Constraint:* Not Null
 - *Description:* Year the first issue in the series was published. It is important to add this to differentiate between series with the same name. It is added as a field for series instead of just using the publication date from the issue table because there is no guarantee that the first issue from every series will be included in a reading order and thus stored in the database.
 - *Example:* 1938
- Field: description
 - *Data Type:* text
 - *Data Format:* “Description paragraph”
 - *Description:* This is the series description pulled from <https://leagueofcomicgeeks.com>.
 - *Example:* “Rocketed to Earth from the doomed planet Krypton, the baby Kal-El was found by a kindly couple and raised as Clark Kent. Developing extraordinary powers thanks to Earth's yellow sun, Clark fights for truth, justice and the American way as Superman, the Man of Steel!”

TABLE: Issue

- Field: id*
- Field: SERIES_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the series the issue is a part of.

*See TABLE: Series

- Field: number
 - *Data Type:* varchar
 - *Description:* Using a string type so that Issue 0 is not counted as a NULL field and so that I can input Annual Issues as “Annual 1.” The number given to the issue. May not necessarily correspond to that issue’s number within the series. For example, the latest issue of Action Comics is #1057 even though the current Action Comics series only launched in 2016 because that series launched it’s first issue as #957, using a legacy numbering system that takes into account all issues in series titled Action Comics.
- Field: publication_date
 - *Data Type:* date
 - *Data Format:* YYYY-MM-DD. If full date is unknown input year and month as YYYY-MM or just the year as YYYY. Standard: [W3CDTF](#).
 - *Constraint:* Not Null
 - *Description:* Date of the first printing of an issue. The exact date may not be available for some older issues which is why the use of just year and month or just year is allowed. If no publication date is available, the year of publication will be the same as the one found on the cover date. Pulled from <https://leagueofcomicgeeks.com>
 - *Example:* 1938-04-18
- Field: cover_date
 - *Data Type:* date
 - *Data Format:* YYYY-MM. If full date is unknown input just the year as YYYY. Standard: [W3CDTF](#).
 - *Constraint:* Not Null
 - *Description:* Date printed on the cover of the issue. It is used to indicate when the issues should be removed from the store stands and is usually dated 2 to 3 months after the publication date. Pulled from <https://leagueofcomicgeeks.com>.
 - *Example:* 1938-06-00
- Field: is_anthology
 - *Data Type:* boolean, tinyint
 - *Data Format:* 1 (True) or 0 (False)
 - *Size:* 1
 - *Constraint:* Not Null
 - *Description:* If the issue is an anthology consisting of more than two stories, then mark as True. Else mark as false. Check stories listed in <https://leagueofcomicgeeks.com>.
- Field: has_backup
 - *Data Type:* boolean, tinyint
 - *Data Format:* 1 (True) or 0 (False)
 - *Size:* 1
 - *Constraint:* Not Null
 - *Description:* If the issue has a main story and a second backup story stories, then mark as True. Else mark as false. Issues where is_anthology is True, should have has_backup as False. Check stories in that issue listed in <https://leagueofcomicgeeks.com>.
- Field: description
 - *Data Type:* text
 - *Data Format:* “Description paragraph”
 - *Description:* This is the description pulled from <https://leagueofcomicgeeks.com>. Often this is an official synopsis given by the publisher.
 - *Example:* “The debut of the world's most recognizable superhero! Learn the original origin of the Man of Steel in his very first appearance! This issue also features the first appearance of Superman's Lois Lane and the superhero magician Zatara!”

TABLE: Story

- Field: id*
- Field: title
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Title Case”
 - *Description:* The title of the story found within the comic book issue. Pulled from <https://leagueofcomicgeeks.com>. Not all stories have titles. If it does not type in “Series Title #N”
 - *Example:* “Superman Origin”
- Field: CONTINUITY_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the continuity the story takes place in.
- Field: CROSSOVER_EVENT_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key
 - *Description:* Unique numeric identifier for the event during which the story takes place. Most stories will not be part of crossover events, and this will be left blank.

TABLE: Has_Story

- Field: ISSUE_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the issue the story is a part of. One story can be printed in multiple issues.
- Field: STORY_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the story printed in the issue. One issue can have multiple stories.

TABLE: Person

- Field: id*
- Field: first_name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Name”
 - *Constraint:* Not Null
 - *Description:* This refers to the first name of a real person involved in the making of a story within a comic issue. Pulled from <https://leagueofcomicgeeks.com>.
 - *Example:* “Jerry”
- Field: last_name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Lastname”
 - *Description:* This refers to the last name of a real person involved in the making of a story within a comic issue. Pulled from <https://leagueofcomicgeeks.com>.
 - *Example:* “Siegel”

TABLE: Role

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Name”
 - *Constraint:* Not Null
 - *Description:* This refers to the role a person took in the making of the story. Roles can include Writer, Artist(for Inker, Penciler, and Colorist), Letterer, Editor, Editor in Chief. Pulled from <https://leagueofcomicgeeks.com> and <https://comicvine.gamespot.com/>.
 - *Example:* “Writer”

TABLE: Contributed To

- Field: id*
- Field: ROLE_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for a role a person has in the story.
- Field: PERSON_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the person who has the role in the story
- Field: STORY_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for a story where the person has that role.

TABLE: Tag

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Title Case”
 - *Constraint:* Not Null
 - *Description:* The name for the tag. Tags are going to be filterable elements found in the description of issues in on reading order sites. The planned tags for now are Character Death, Character Resurrection, Retcon (previously established history is overwritten),Character Retcon (a character’s history is overwritten), Character Altering Event (for events that have significant repercussions for characters later on),Character Reintroduction, Marriage, Relationship Event, Team Event (for events that alter the status quo of a team), Concept Introduction (e.g., first appearance of the Batcave). I want to leave open the possibility of adding more as I go through the reading orders and the issues themselves. I’m tying this field to the issue rather than the story because the notes in reading orders do not usually specify the story.
 - *Example:* “Character Altering Event”
- Field: description
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”
 - *Description:* This refers to the description of the tag
 - *Constraint:* Not Null
 - *Example:* “Character Altering Event refers to the moments in a character history which significantly impacts the trajectory of a character but is not covered by other existing tags such as Character Death, and Character Resurrection”

*See TABLE: Series

- Field: type
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Type”
 - *Description:* This refers to the tag’s subtype. Options are Character, Team, Relationship. Can also be left blank as not every tag has to belong to a subtype.
 - *Example:* “Character”

TABLE: DC_Character_Tag

- Field: id*
- Field: TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the tag. Character tag applies to character specific tags like Character Death, Character Reintroduction, Character Resurrection, Character Retcon, Character Altering Event.
- Field: DC_CHARACTER_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for character the tag is being applied to.

TABLE: Team_Tag

- Field: id*
- Field: TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the tag. Team tags applies to team specific tags like Team Event.
- Field: TEAM_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the team the tag is being applied to. E.g., Id 1 = Justice League event.

TABLE: Relationship_Tag

- Field: id*
- Field: TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the tag. Relationship tag applies to relationship specific tags like Relationship Event.
- Field: RELATIONSHIP_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the relationship the tag is being applied to. E.g., Id 1 = Clark Kent and Lois Lane romantic relationship event.

TABLE: Has_Tag

- Field: id*
- Field: ISSUE_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the issue. One tag can be present in many issues.

- Field: TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the tag for tags that do not belong to a type. One issue can have many tags.
- Field: DC_CHARACTER_TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* When the tag in the tag_id field is a character_tag, the the specific id for that tag as it applies to a specific character is added. Ex: if tag_id makes reference to the tag character death in an issue where superman dies, then DC_character_tag_id for superman character death is added.
- Field: TEAM_TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* When the tag in the tag_id field is a team_tag, the specific id for that tag as it applies to a specific team is added.
- Field: RELATIONSHIP_TAG_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* When the tag in the tag_id field is a relationship_tag, the specific id for that tag as it applies to a specific relationship is added.
- Field: note
 - *Data Type:* varchar
 - *Data Format:* "Description paragraph"
 - *Size:* 1- 65,535.
 - *Description:* This refers to a note explaining why the tag is added.
 - *Example:* "Barabra Gordon is shot and paralyzed from the waist down"

TABLE: DC_Character

- Field: id*
- Field: first_name
 - *Data Type:* varchar
 - *Data Format:* "Name"
 - *Constraint:* Not Null
 - *Description:* A character's first name. Use the name they use most often. E.g., Use Clark instead of Kal for Superman, but Kara instead of Linda for Supergirl. May not always correspond to their legal first name. E.g., Use Dick instead of Richard for Dick Grayson. This will usually match the names the character is referred as in the reading order and in <https://leagueofcomicgeeks.com>.
 - *Example:* "Clark"
- Field: last_name
 - *Data Type:* varchar
 - *Data Format:* "Lastname"
 - *Size:* 1- 65,535.
 - *Description:* A character's last name. In cases where a character is introduced with a last name that changes much later in their history due to marriage or adoption, keep the last name with which they were introduced with and add the new last name as an alias (e.g., Drake not Wayne for Tim Drake). In cases a character has more than one name use the last name use the one they use most often (e.g., Use Kent instead of El for Superman, but Zor-El instead of Lee/Lang/Danvers for Supergirl). This will usually match the names the character is referred as in the reading order and in <https://leagueofcomicgeeks.com>. Some characters have no known name, name and this field can be left blank.

*See TABLE: Series

- *Example:* “Kent”
- Field: description
 - *Data Type:* text
 - *Data Format:* “Description paragraph”
 - *Description:* This is the character description pulled from <https://comicvine.gamespot.com/>.
 - *Example:* “Sent to Earth as an infant from the dying planet Krypton, Kal-El was adopted by the loving Kent family and raised in America's heartland as Clark Kent. Using his immense solar-fueled powers, he became Superman to defend mankind against all manner of threats while championing truth, justice, and the American way!”

TABLE: Alias

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Title Case”
 - *Constraint:* Not Null
 - *Description:* Alternate names that a character is known as, separate from their superhero/villain mantle (Kal-El is an alias for Clark Kent but Superman is a mantle). The data will be pulled from <https://comicvine.gamespot.com/> and <https://leagueofcomicgeeks.com> but not all of the aliases listed on those sites match the definition I am using (E.g., Comic Vine has Clark Kent listed as an alias and LeagueofComicGeeks has Superman listed as an alias) so it needs to be evaluated on a case by case basis.
 - *Example:* “Kal-El”
- Field: DC_CHARACTER_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key
 - *Description:* Unique numeric identifier for the character the alias is associated with.
- Field: notes
 - *Data Type:* varchar
 - *Data Format:* “description paragraph”
 - *Description:* This is a short paragraph explaining the nature of the Alias. This is not pulled from anywhere and will have to be written in by me based on prior knowledge or research.
 - *Example:* “Kal-El is the Kryptonian name given to Clark Kent by his birth parents”

TABLE: Mantle

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Data Format:* “Name”
 - *Size:* 1 – 500
 - *Constraint:* Not Null
 - *Description:* A Superhero or Villain name a character has been known as. The data will be pulled from <https://comicvine.gamespot.com/> and <https://leagueofcomicgeeks.com> from the Aliases or Super Name fields. These do not follow the same definition as the one I use so it needs to be evaluated on a case-by-case basis. Multiple characters can hold the same mantle, sometimes even at the same time.
 - *Example:* “Superman”
- Field: description
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”

*See TABLE: Series

- *Description:* This is a short paragraph explaining the nature of the mantle. This is not pulled from anywhere and will have to be written in by me based on prior knowledge or research.
- *Example:* “Usually associated with Clark Kent this mantle refers to the main protector of Metropolis”

TABLE: Has_Mantle

- Field: id*
- Field: MANTLE_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for a mantle a character has
- Field: DC_CHARACTER_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for a character that has a mantle.
- Field: note
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”
 - *Description:* This is a short description of the character’s relationship to the mantle. Not pulled from any data source but based on my knowledge or research.
 - *Example:* “Barbara Gordon was the second Batgirl. She quit being Batgirl shortly before becoming paraplegic. She became Batgirl once more during the New 52 relaunch having regained the use of her legs due to a neural implant.”
- Field: current
 - *Data Type:* boolean, tinyint
 - *Data Format:* 1 (True) or 0 (False)
 - *Constraint:* Not Null
 - *Description:* If the character as of their most recent appearance still has that mantle mark as 1 (True). Otherwise mark as 0 (False). A character’s most recent appearance can be searched in <https://leagueofcomicgeeks.com>.

TABLE: Team

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Name”
 - *Constraint:* Not Null
 - *Description:* This refers to the name of a team composed multiple heroes or villains. This team can be a formal team that is given a name and base of operations in the comics such as the Justice League, or it can be an informal recurrent association of characters such as the Super-Family.
 - *Example :* “Justice League”
- Field: description
 - *Data Type:* text
 - *Data Format:* “Description paragraph”
 - *Description:* This is the description of the team pulled from <https://leagueofcomicgeeks.com>.
 - *Example:* “For decades, the Justice League has saved humanity from the worst threats it has ever faced, from alien warlords to ancient demons and powerful sorcerers, to the God of Evil himself, Darkseid. And every time, they have overcome the odds and saved the human race.”

TABLE: In_Team

- Field: id*
- Field: TEAM_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for a team
- Field: DC_CHARACTER_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for a character in that team
- Field: note
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”
 - *Description:* This is a short description of the character’s relationship to a team. Not pulled from any data source but based on my knowledge or research.
 - *Example:* “Dick Grayson was a founding member of the Teen Titans”
- Field: current
 - *Data Type:* boolean, tinyint
 - *Data Format:* 1 (True) or 0 (False)
 - *Size:* 1
 - *Constraint:* Not Null
 - *Description:* If the character, as of their most recent appearance, is still a member of the team mark as 0 (True). Otherwise mark as 1 (False). Pulled from <https://leagueofcomicgeeks.com>.

TABLE: Relationship_Type

- Field: id*
- Field: type
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Type”
 - *Constraint:* Not Null
 - *Description:* The type of relationship. For now, the types are Family, Romance, Mentorship, Friendship, Antagonism. All except Family can apply to one-sided relationships and two characters can have more than one relationship type.
 - *Example:* “Romance”
- Field: description
 - *Data Type:* varchar
 - *Data Format:* “description paragraph”
 - *Description:* This is a short paragraph explaining the definition of the relationship type.
 - *Example:* “Romance applies to relationships in which characters have expressed romantic or sexual interest in each other.”

TABLE: Relationship

- Field: id*
- Field: DC_CHARACTER_id_1
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for one of the characters in a relationship
- Field: DC_CHARACTER_id_2
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null

- *Description:* Unique numeric identifier for the other character in that relationship.
- Field: RELATIONSHIP_TYPE_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the relationship type from the relationship_type table.
- Field: current
 - *Data Type:* boolean, tinyint
 - *Data Format:* 1 (True) or 0 (False)
 - *Size:* 1
 - *Constraint:* Not Null
 - *Description:* If the characters as of their most recent appearance still have that relationship the team mark as 0 (True). Otherwise mark as False. Not pulled from any data source but based on my knowledge or research.
- Field: note
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”
 - *Description:* This is a short description of the characters relationship. Not pulled from any of the data sources but based on my knowledge or research.
 - *Example:* “Clark and Lois have been romantically involved since the creation of both characters. They are currently married.”

Note: Though not in the original scope of the database I realized that a lot of the notes in the reading order list highlighted relationship specific events so I decided to keep track of relationships so I could add the relationship tags to issues.

TABLE: Continuity

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Name”
 - *Constraint:* Not Null
 - *Description:* DC has multiple continuities; some take place on alternate Earth’s or timelines. This database focuses on only main Earth and Main timeline stories, however the main earth and timeline have been rewritten multiple times, resulting in many different continuities within it. Pre-Crisis Continuity consists of comics published before Crisis on Infinite Earths. Post-Crisis consists of comics after Crisis on Infinite Earths but before Flashpoint. The New 52 consist of comics post Flashpoint but before Rebirth. After Rebirth there is some disagreement amongst the three reading order websites as to whether the Infinite Frontier or Dawn of DC initiatives should be considered different continuities or part of Rebirth. I have decided based on my own experience reading comics, to use Pre-Crisis, Post-Crisis and then New-52, Rebirth, and Infinite Frontier and Beyond. This is because New 52, Rebirth and Infinite Frontier feel distinct in their approaches to integrating Pre-Flashpoint continuity, but Dawn of DC feels more like a continuation of Infinite Frontier than a proper reboot.
 - *Example:* “Post-Crisis”
- Field: description
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”
 - *Description:* This is the description of each continuity. It will be written by me synthesizing information from <https://comicbookreadingorders.com/dc>, <https://www.comicbookherald.com/where-to-start-with-dc-comics>, and <https://www.comicbooktreasury.com/crisis-dc-comics-reading-order/>

*See TABLE: Series

- *Example:* “The post-crisis continuity starts after the crossover event Crisis on Infinite Earths in 1985 which rewrites the history of the DC universe with all characters given new or updated origins, but before the Flashpoint event in 2011 which rebooted the DC Universe once more”

TABLE: Crossover_Event

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Data Format:* “Name”
 - *Size:* 1 – 500
 - *Constraint:* Not Null
 - *Description:* An Event in comic book terminology refers to a storyline that crosses over multiple series and may have some lasting consequence to the characters involved. They will be given names by the publisher.
 - *Example:* “Crisis on Infinite Earths”
- Field: description
 - *Data Type:* varchar
 - *Data Format:* “Description paragraph”
 - *Description:* This is the description of each event will be written by me synthesizing information from description paragraphs found in <https://comicbookreadingorders.com/dc>, <https://www.comicbookherald.com/where-to-start-with-dc-comics>, and <https://www.comicbooktreasury.com/crisis-dc-comics-reading-order/>
 - *Example:* “Crisis on Infinite Earths in 1985 was company wide crossover event. This was a multiversal catastrophe which destroyed multiple parallel universes and recreated the main universe.”

TABLE: Features

- Field: id*
- Field: STORY_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for one of the stories in an issue.
- Field: DC_CHARACTER_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key. Not Null
 - *Description:* Unique numeric identifier for one of the characters in an issue.
- Field: HAS_MANTLE_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key
 - *Description:* Unique numeric identifier for one a character using a mantle. The reason why both this id and the character id are here is that some characters can appear in an issue outside of a mantle or they might be civilian characters who do not have a mantle.
- Field: IN_TEAM_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key
 - *Description:* Unique numeric identifier for a character appearing as a member of a team.

TABLE: Reading_Order

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Data Format:* “Reading Order : Source”
 - *Size:* 1 – 500

- *Constraint:* Not Null
- *Description:* This the name given to the reading order. Mostly used for reference. The source and character/event/team/continuity.
- *Example:* “Barbara Gordon: CBRO”

- Field: SOURCE_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the source the reading order was obtained from.

- Field: type
 - *Data Type:* varchar
 - *Data Format:* “Type”
 - *Size:* 1 – 500
 - *Constraint:* Not Null
 - *Description:* This refers to the reading order subtype. Options are Character, Continuity, Event, Team. These categories can be found in the headings of the reading order websites. Reading orders must belong to one type.
 - *Example:* “Character”

- Field: link
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “https://www.source.com/readingorder”
 - *Description:* The link to the reading order
 - *Example:* <https://www.comicbooktreasury.com/crisis-dc-comics-reading-order/>

TABLE: DC_Character_Reading_Order

- Field: READING_ORDER_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the reading order. Character subtype applies to character specific reading orders. The sites all have separate reading order lists for specific characters. The name of the character can be found in the title of the reading order.

- Field: DC_CHARACTER_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the character the reading order is about.

TABLE: Continuity_Reading_Order

- Field: READING_ORDER_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the reading order. In the general dc reading orders found on the sites, the specific continuity will usually be in the heading for a subsection.

- Field: CONTINITY_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the continuity the reading order is about.

TABLE: Crossover_Event_Reading_Order

- Field: READING_ORDER_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the reading order. The sites all have separate reading order lists for specific events. The even can be found in the title of the reading order.
- Field: CROSSOVER_EVENT_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the event the reading order is about.

TABLE: Team_Reading_Order

- Field: READING_ORDER_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the reading order. The sites all have separate reading order lists for certain teams, specially when the team members rarely appear in other content outside of the team such as the Doom Patrol. They are usually listed along side the character reading orders and do not have their own website section.
- Field: TEAM_id
 - *Data Type:* Integer
 - *Constraint:* Foreign Key, Not Null
 - *Description:* Unique numeric identifier for the team the reading order is about.

TABLE: Source

- Field: id*
- Field: name
 - *Data Type:* varchar
 - *Size:* 1 – 500
 - *Data Format:* “Title Case”
 - *Constraint:* Not Null
 - *Description:* The source for the reading orders. This will be the name of the website. For now, the only sources will be Comic Book Herald, Comic Book Reading Orders, Comic Book Treasury
 - *Example:* “Comic Book Treasury”
- Field: description
 - *Data Type:* text
 - *Data Format:* “Description paragraph”
 - *Description:* small description of the source pulled from the About sections of each site.
 - *Example:* “Welcome to Comic Book Treasury, a blog dedicated to comics reading order for Marvel, DC and other publishers. We cover major characters and other favorites as well as events, and more!”
- Field: link
 - *Data Type:* varchar
 - *Data Format:* “https://www.readingordersource.com”
 - *Description:* The link to the homepage of each site
 - *Example:* <https://www.comicbooktreasury.com/>

TABLE: In_ReadingOrder

- Field: id*
- Field: READING_ORDER_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null

*See TABLE: Series

- *Description:* Unique numeric identifier for the reading order. One reading order has many issues.
- Field:ISSUE_id
 - *Data Type:* Integer
 - *Constraint:* Primary Key, Foreign Key, Not Null
 - *Description:* Unique numeric identifier for an issue in a reading order. One Issue can be in many reading orders.
- Field:position
 - *Data Type:* Integer
 - *Constraint:* Not Null
 - *Description:* Indicates the position of an issue in the reading order as a number (e.g. 1st = 1).

Note: The mapping of the relationships between characters, teams, continuities, and events reading orders and issues might seem odd because I am, for example, mapping what characters are in an issue but also if that issue is in that character's reading orders. These are distinct concepts. An issue in the database might, for example, feature Barbara Gordon but not be in any of her reading orders because it is inconsequential for her character.