Code	Description	Example
AI, ML, GNNs use case	The KG is being used for training AI/ML/GNN models, or, AI/ML/GNNs are being used to understand KGs	PID15: I was going to build this sort of tool for all the machine learning engineers and data scientists to use
Baseball card	The participant has found success or wants to summarize KG analysis into a small set of disgestible information, presented as a knowledge card or "baseball card"	PID14: Yeah. So probably the easiest way to talk about them is, you know, when you Google something, it is a little box that pops up on the right hand side that has, **like a bunch of relevant information, all in one place and sort of an organized format. That's a knowledge card.**
Inherent KG differences	Some inherent difference about one or more KGs, typically due to the way they store data, structure data, allow querying of data, etc.	PID1: Visualizing really large KGs is really difficult So having an effective way to drill down into what you care about It depends on what type of KG you're looking at
KG challenge, data provenance	Creating and curating KGs is hard because of data provenance challenges, including incorrect, missing, or obselete input data	PID18: The accuracy of the data is difficult because you have the same data flowing into multiple systems for no reason at all
KG challenge, difficult to aggregate data	A challenge of using KGs is aggregating or summarizing the data found in a KG. Whether to present that data to end-users, or to use that data in model training, testing, or specific use cases.	PID18: There's tons of aggregations that just get lost in translation or historical systems that got outdated It shouldn't be a manual effort where you need to document this for eternity, but you should have a system that can automatically pull out these connections Even in pharma, it's difficult to pull out health records, which are different from other data in pharma development
KG challenge, end user trust in KG systems	End users do not understand and trust systems using KGs, if the underlying KG is exposed to them	PID17: But you know, obviously based on your interest to the visualization is a a major challenge. And in our experience the more we can shelter the end user from the underlying structure the better their willingness to interact with the data and and accept the results that come out of it. **As soon as the level of complexity of the graph reaches a certain level on the screen, they really tend to shut down and and not trust any of it**. I think there's just, there's just too many options and and too much to think through. I think the the cognitive load just gets so high that they they tend to just you know, close it off and say, there's there's too much here. There's no way to really prioritize these things. It doesn't it doesn't help.
KG challenge, entity disambiguation	One challenge with KGs is determining whether two entites are actually different or the same	PID8: One problem is just entity disambiguation. We don't do a good job of that on the ingestion side. We take the data as it is. So you're not actually sure that two nodes that are different are actually different.
KG challenge, path discovery chokepoints	When doing path discovery, one challenge is that there are certain chokepoints in the KG that lead to many irrelvant paths	PID8: The weakness layer is very small and they're connected to a lot of nodes. So if you kind of do that two step linkage between the two nodes, you end up with probably a lot of things that are irrelevant and more nodes than you actually want.
KG challenge, scalability	Creating and curating (or exploring, mining insights from) KGs is hard because of of the sheer volume of data necessary to process	PID4: Yeah, I would agree, like most of the time, we would only want to see like a small small snapshot because of the performance issues. Because otherwise your system just freezes.
KG challenge, schemas	A challenge of using KGs is not knowing the schema for a node, or not knowing when a schema for a particular kind of node is consistent across the entire KG	PID6: Matt and Miroslav. They both have a type of Person. Is it true that all of those green boxes, whatever they are, have a type and a birthday? **Is it true that they have like exactly one birthday? You know, like that's never present unless you're looking at like some schema. I would want to see documentation for things like the start dates and the end dates.** That's kind of a red flag for me, right? Really that's probably an interval. And if it's a start date is like an open interval or a closed interval.
KG creater	The participant has experience creating/curating KGs	PID5: So you can timestamp when nodes are created, when connections are created
KG curation uses closed source data	The participant uses a KG that was in part created using non-public data, usually internal to the company	PID15: So like you know, if if we have a a drug or something that there's a clinical trial for, I want to be able to just very quickly like, have, like all the interesting properties about that drug on a page
KG curation uses open source data	The participant uses a KG that was in part created using publically available data	PID15: But Wiki data was a knowledge graph, but it was really exciting to use right, because Wikipedia is kind of like the collection of all human knowledge

Code	Description	Example
KG end user	The participant has experience as the end user of a system powered by a KG	P18: I didn't create the graph myself. I was part of the team that worked on it, but mostly I was using what they already made
KG used for data provenance	Participant has used KGs to manage enterprise data to be able to 1) Keep data consistently in source locations 2) Understand where data is located 3) Making querying data easier	PID14: So all I have to do is query the graph, and I can implicitly query every single source system across my company. Which becomes really powerful. Because, hey maybe I don't just want data from one system. Maybe I want data from like 5 systems, and I want it at the same time in the same place. And so now I can write a query that lets me do that, and it's going to virtualize that data from 5 different sources of data without creating copies of them. And so that whole concept is something that in the last 2 years people started wanting to call a data fabric. So that's what that language is about.
KG used for input to other downstream analysis	The KG is not the focus of the overall analysis per se, but rather one of several inputs fed into a downstream analysis pipeline or model	PID9: The direction we've moved now is using the knowledge graph as a pure source of data. So given a system where you're asking the question of, okay, what is anomalous here? You might have several sources of data.
KG used for knowledge base	Participant has used KGs to collect academic / enterprise / public knowledge in one location	PID16: So one was like a a knowledge graph that was -- it actually had like billions of nodes. And **it was scraped from scientific papers.** ... And I guess the other thing is like the with the the knowledge graph that's like scientific papers and everything like, the easy and fast part is the important part, because, like there's so many papers out there. **I don't have to read like 10 papers to find out a piece of information.**
KG used for node and or link classification	Participant uses the KG to try and predict classes to nodes and or links	PID9: What we were trying to do was use data that did not contain anomalies, but did contain information about just like context. So for example, fire hydrants are normally red. Chainsaws are normally in a shed. And our hope was basically using that kind of like source of context, we could do anomaly prediction for things like should I, should a chainsaw be in the kitchen? The answer is no here.
KG used for node and or link prediction	Participant uses the KG to try and predict missing nodes and or links	PID12: We are looking at KG embeddings so say that we would expect this link to be there
KG used for node and or link regression	Participant uses the KG to try and predict a value (discrete or continuous) for nodes or links	PID15: So, like the classic example was like, if you're trying to make a prediction for countries like you know. What is the country's total GDP going to be next year or something.
KG used for path discovery	Participant uses KG for path discovery, ie to find (often long and previously unknown) paths between nodes in the graph	PID17: Yes, well, definitely, pathways is a big open-ended question, right? We have pretty rough definitions of some pathways and a lot of overlap between a lot of pathways. And and so it gets pretty messy pretty quickly if you try and do something like, just look at geneset based associations. **So we tend to prefer the graph oriented approach to looking at pathways** because it allows us a another level of of clustering and and community assignment to to help really point out areas that are more significantly associated with whatever perturbation we're looking at, whether it's a a mutation, or a knockout, or or a chemical treatment...It's mostly when we don't see a good, strong individual signal, and we only see in combination with, several different entities that we see a strong association. Then we'll go to a a graph level to try and understand what those shared associations are.
KG used for question answering	KG is used or can be used for question-answering, either automatically or manually	PID15: Like, usually I'd have like something specific I'd wanted to have answered when I would go to the knowledge graph.
KG used for saving time	Participant uses the KG to analyze a lot of data quickly, which saves a lot of time	PID17: Basically what we use them for is as a way to help inform our otherwise you know subjective experience with literature or interpretations around multiple internal experimental results. I think **the most clear value we have seen so far is one largely the time savings** that we get by looking across relationships and really helping us sort of filter down the massive amount of literature and and focus in into particular areas.

Code	Description	Example
KG Visualization Opportunity or Suggestion	There is an opportunity to leverage visualization for KGs, either a design suggestion or system/tool suggestion, etc. that is different than what is the standard approach (gephi, node link diagrams, baseball cards)	PID14: I think Gephi right now has this...I don't think you can do like 3D rotation and something. I don't know if you use the tensor projector by Google or by Tensorflow
Meta, confusing, not sure	The coder is not sure about something	
Meta, possible quote	The participant's response may be a good quote for the paper	
No visual tools for KG creation	The participant KG creater does not use visual tools when creating or curating KGs	PID8: I actually don't use anything to visualize them.
No visual tools for KG end user	End users of the KG analysis system do not use visual tools	PID16: And so we we found that trying to basically hide the graph from the user and just give them summarized sets of of information that have been basically extracted from the graph tends to increase their their appetite for interacting with it.
Node-link diagrams are difficult to interpret at scale by end users (cluttered design, poor layout design)	When there are many nodes and links, node link diagrams are difficult to interpret by KG creators and end users	PID17: I just want to go see it. Why? Just because I want to go see it. I hear this all the time, and then you see something small like, okay, You saw it small. So what, I don't know. I saw it. Show me something bigger. It turns into a hairball. I can't deal with that. And it's like, okay. So, then, this is the story story of graph visualizations all the time.
Node-link diagrams are good eye candy	Node-link diagrams are attractive to look at in presentations and demos	PID13: Again, I will argue that it makes great slide decoration.
Node-link diagrams have performance issues at scale	When there are many nodes and links, the decreased computer performance leads to usability challenges	PID4: like most of the time, we would only want to see like a small small snapshot because of the performance issues. Because otherwise your system just freezes.
Querying is difficult	Using KG querying languages is difficult	PID15: writing the queries myself. And I didn't find a lot of things that made it easy
Social-technical challenge	The challenges of using KGs in an enterprise environment are people, communication, teamwork, and organization structure related (ie social) moreso than technical	PID18: But I would think that right now the challenges are mainly social challenges, not as much technical challenges. It's about understanding, who are the people who has knowledge? Where do you get this stuff out of? Basically, I think I always say it, it's almost an AI complete problem. Not even the humans agree what things mean. How do how do we expect that a machine is going to come up with the right answer? So a lot of this is more of a people process challenge. Like, how do you, how do you create a knowledge graph? How do you create this ontology? **It's a bit more of a people and process than a technology. I would argue that the technologies are there. We don't need more technology. We just need to focus on the people in the process. It's a it's a social-technical phenomena.** And we've been focusing on the on the technical phenomena up to now, and we need to change. And that's the challenge that we have not focused on the social side. I could go deeper, I can go, I could spend an hour on each of these questions.
Specific tool(s) used	Reference to a tool for KGs: visualization tools, storage tools, data science / analysis tools, etc.	"If it's a large enough graph we use Neo4j to store it and query it because, the CYPHER language makes a lot of that really simple and straightforward, and it has a lot of good data science tools already built in. For more direct interaction with the graph we tend to program everything in Python using NetworkX or iGraph to extract layers and interact with those. And if we need to do sort of a canned visualization with some sort of independent querying, we'll tend to use Gephi or something at the desktop level." PID16
Text-driven or NLP use case for KGs	Someone is using a KG for an NLP use case (prediction, learning, analysis), for understanding or predicting text, for storing text for a broader NLP use case, etc.	PID12: Right now we are trying to use NLP techniques to try to identify causal relationships. Not only co-occurring, but that there are statements about how they are connected
Usage of adjacency matrix	The participant uses an adjacency matrix to represent their KG (vs some other form, like Neo4J)	PID11: So I just created the adjacency matrix and feed that into the models
Usage of Gephi	The participant has used Gephi for visualization in their experience with KGs	PID54: I think one thing that was kind of cool with Gephi is that like you can kind of see over like you can drag a time bar underneath

Code	Description	Example
Usage of Neo4J	The participant has used the Neo4J graph database in their experience with KGs	PID14: For property graphs, the sort of like big 200lb gorilla in that market is Neo4J right.
Usage of NetworkX	The participant has used NetworkX in their experience with KGs	PID5: On the [PROGRAM_1], just NetworkX pricing and drawing. Not ideal. Not great at all, especially when ou have directed edges. On [PROGRAM_2], also NetworkX just for debugging purposes
Usage of node-link diagram for sanity checking	The participant uses network diagrams exclusively for "sanity checks", by visualizing subsets of the generated KG, to make sure that the KG creation proces is working as intended	PID14: No, I don't use it along the whole workflow, so I usually use it just as a way to make sure, like to try to see if nothing weird is going on or to see whether the node, whether the graph is connected. Or if I want to do some kind of filtering on the graph which Gephi he does allow allow me to do.
Usage of RDF	The participant has used the Resource Description Framework in their experience with KGs	PID14: RDF is, they're triples basically. The storage unit for them is: here's object. Here is predicate object but...It's been a minute. **It's resource description framework.** It's based on a worldwide web consortium standard. so it was something that came out of the semantic web projects in the early part of worldwide web stuff with Tim Burners Lee (sp?). So if you sort of look at the history of knowledge graphs, they trace their roots to semantic web technology. Right? Okay, Well, let's move from what we had structured the the web as before, to something that was looking at semantic connections between things. And then knowledge graph as a term really starts getting used first by Google. They kind of make that the go-to term instead of semantic web.
Usage of SQL	The participant has used a SQL database in their experience with KGs	PID8: It's actually stored in a SQL database
Visualization challenges for KGs are domain specific for end users	The participant believes that visualization challenges are domain specific and not generalizable, ie that creating a useful visual system depends on what the end user is looking for	PID18: But then you have something very specific. Oh, I am looking for this very specific thing. So so it's kind of being very specific and very general. And I think we, the Googles of the world right? It's like. Oh, you think about everything as search. And and then you want, and it's a very general thing. But in reality, we have some intentions about things like you don't just show up to amazon.com and like. Let me just go click around. I mean, okay, maybe people do. But why do they do that then, right? Where do they end up right? So this is about understanding what the intentions are. So basically, what are the use cases? So that's why, I think that the premise of that question is is, is is is incorrect. Author 1: So would you say, then it's, it's just very **domain specific**, depending on the user, depending on the use case? PID18: Yeah, sure, exactly
Wikipedia is good exploration and discovery tool	The participant likes to use Wikipedia as a vehicle to explore the underlying graph data	PID14: Is there a way to like maintain that discoverability and like an example of something that is not a perfect one to one relationship but I think does this fairly well, and is maybe a starting point to think about this is Wikipedia. Right? I look up an article on **Wikipedia, and they use hyperlinks to basically connect me to things that are related to it. So you could visualize Wikipedia as a graph** like plenty of people do it because they're all involved in the same like open data spaces. But you have oh like the the Wikipedia game, right? You can get it from anywhere in Wikipedia to anywhere else in like 6 clicks. It's a highly dense network absolutely like a graph. But do I explore that graph by looking at a graph? No, I explore it by looking at static pages with links to the things that are related to it. And so is there a way to have a similar like card hopping approach? In some domains for some problems, maybe.