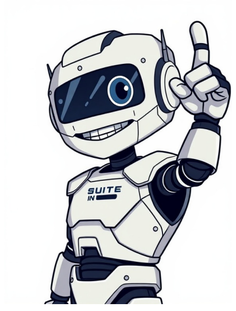
Click to prove you're human



You can’t perform that action at this time. Placeholder data allows a query to behave as if it already has data, similar to the initialData option, but the data is not persisted to the cache. This comes in handy for situations where you have enough partial (or fake) data to render the query successfully while the actual data is fetched in the background. Example: An individual blog post query could pull "preview" data from a parent list of blog posts that only include title and a small snippet of the post body. You would not want to persist this partial data to the query result of the individual query, but it is useful for showing the content layout as quickly as possible while the actual query finishes to fetch the entire object. You can supply placeholder data for a query to cache before you need it in several ways. One approach is to pass the placeholder data as a value when setting up the query. For instance, `const result = useQuery('todos', () => fetch('/todos'), { placeholderData: placeholderTodos })`. This method allows you to quickly show the content layout while the actual query finishes fetching the entire object. If accessing a query's placeholder data is resource-intensive or not something you want to perform on every render, you can pass a function as the `placeholderData` value instead. This function will be executed only once when the query is placeholderized, saving memory and CPU resources: `const result = useQuery('todos', () => fetch('/todos'), { placeholderData: () => generateFakeTodos() })`. In some cases, you may be able to provide the placeholder data for a query from the cached result of another. For example, searching the cached data from a blog post list query for a preview version of the post and then using that as the placeholder data for your individual post query: `const result = useQuery(['blogPost', blogPostId], () => fetch('/blogPosts'), { placeholderData: () => queryClient.getQueryData('blogPosts').find(d => d.id === blogPostId) })`. These methods enable you to efficiently manage placeholder data and improve the user experience by displaying content quickly, even before the actual query finishes fetching all the necessary information.To balance a chemical equation, start by assigning variables to the coefficients of each element. Then, write down equations based on atom conservation and solve the system. For example, C2H6 + O2 = CO2 + H2O: a C2H6 + b O2 = c CO2 + d H2O Write down equations based on atom conservation: 2 a = c 6 a = 2c a = 2d b = 2c + d Assign one of the coefficients to 1 and solve the system. For instance, a = 1 and c = 2. a = 1 c = 2 d = 6 Adjust coefficient to make sure all of them are integers. Multiply all coefficient by 2 to arrive at the balanced equation with integer coefficients: 2 C2H6 + 7 O2 = 4 CO2 + 6 H2O Balancing with oxidation number method is useful for redox reactions where electron transfer occurs. Process: identify the oxidation numbers, determine the changes in oxidation state, balance the atoms that change their oxidation state, and then balance the remaining atoms and charges. For example, Ca + P = Ca3P2: Assign oxidation numbers: Calcium (Ca) has an oxidation number of 0 in its elemental form. Phosphorus (P) also has an oxidation number of 0 in its elemental form. In Ca3P2, calcium has an oxidation number of +2, and phosphorus has an oxidation number of -3. Identify the changes in oxidation numbers: Calcium goes from 0 to +2, losing 2 electrons (oxidation). Phosphorus goes from 0 to -3, gaining 3 electrons (reduction). Balance the changes using electrons: Multiply the number of calcium atoms by 3 and the number of phosphorus atoms by 2. Write the balanced equation: Ca + P = Ca3P2 Balancing with ion-electron half-reaction method is best for complex redox reactions, especially in acidic or basic solutions. Process: split the reaction into two half-reactions, balance the atoms and charges in each half-reaction, and then combine the half-reactions, ensuring that electrons are balanced. For example, Cu + HNO3 = Cu(NO3)2 + NO2 + H2O: Learn to balance chemical equations by practicing what you learned.