

Bericht: Cursor

Matyukhina, Elina, 20241842

Im Rahmen des Moduls Datenbanken verwendete ich Cursor als KI-gestützte Entwicklungsumgebung.

Wie half Cursor beim Lernen?

Der zentrale Aspekt war die Unterstützung beim Verstehen von Zusammenhängen. In den Vorlesungen wurden viele theoretische Konzepte behandelt. Bei der praktischen Umsetzung dieses Wissens wurde mir klar, in welchen Bereichen des Lernstoffs noch Lücken bestanden, insbesondere bei komplexeren Modellierungsentscheidungen. Durch gezielte zahlreiche Nachfragen an die KI, konnten meine Verständnisprobleme direkt geklärt werden, wodurch ich zeiteffizient lernte und die Theorie aktiv anwendete, ohne den Arbeitsfluss zu unterbrechen.

Überdies war das Lernen aus den Fehlern wertvoll. Wenn die angewendeten Modellierungsansätze oder der Programmiercode, bezogen auf Syntax oder Semantik, nicht funktionierten, bekam ich sofort Unterstützung von der KI. Mir wurden beim Debuggen nicht nur die Fehler angezeigt, sondern auch alternative Lösungsvorschläge, nach deren Erklärung man sich wieder an den Assistenten wenden konnte. Diese sofortige Rückmeldung über Probleme im Quelltext half, die gleichen Fehler in der Zukunft zu vermeiden.

Auch das Stoßen auf neue Ideen wurde durch KI ermöglicht, um die eigene Herangehensweise zu überdenken und zu optimieren. Bei der zukünftigen Arbeit mit Datenbanken ist es auch entscheidend, da man dadurch die allgemeinen Prinzipien und Muster für die Lösung ähnlicher Probleme lernt.

Zusammenfassend lässt sich sagen, dass der Lernprozess nicht vom Cursor ersetzt wurde, sondern sinnvoll ergänzt.

Stärken

Eine der größten Stärken von Cursor ist die nahtlose Integration der KI in die Entwicklungsumgebung. Im Vergleich zu der separaten Nutzung von üblichen Code-Editors und externen KI, vermeidet man das ständige Wechseln von einem Tool zum anderen. Die Fragen können direkt in Cursor gestellt werden. Darüber hinaus ist die Fähigkeit überzeugend, den Quelltext kontextbezogen zu analysieren. Der KI-Assistent erkennt den aktuellen relevanten Code und passt die Antworten entsprechend darauf. Die oben beschriebenen Faktoren sind für die Beschleunigung des Programmierens entscheidend. Auch die automatische Vervollständigung mittels einer Tabulator-Taste hat die Geschwindigkeit deutlich erhöht.

Eine weitere Stärke ist die Erklärqualität. Cursor kann nicht nur den Code generieren, sondern auch begründen, warum bestimmte Entscheidungen sinnvoll oder notwendig sind. Für die lernende Person ist es somit hilfreich, da nicht nur auf das Ergebnis fokussiert wird, sondern auch auf das tiefere Verständnis.

Des Weiteren bietet Cursor eine Auswahl an Suchmodellen und Assistent-Typen.

Außerdem kann man schnell randomisierte Beispieldaten erstellen, um zu prüfen, ob und wie die SQL-Abfragen funktionieren.

Schwächen

Trotz vieler Vorteile hat Cursor meiner Meinung nach eine Schwäche, die man trotzdem teilweise umgehen kann. Zu dieser Schwäche zählt das Erhalten von nicht immer perfekten Antworten und Lösungsvorschläge, die gerade bei sehr spezifischen Aufgabenstellungen oder ungewöhnlichen Modellierungsentscheidungen etwas ungenau oder verallgemeinert sein können.

Typische Use-Cases

- Intelligente Code-Generierung;
- Debugging & Fehlerbehebung;
- Code-Optimierung;
- Verständnis und Navigation im Codebestand;
- Autovervollständigung und prädiktive Vorschläge;
- Lernen und Dokumentation;
- Anpassung und Integration.

Empfehlungen für kommende Jahrgänge

Die Nutzung von Cursor ist klar zu empfehlen, unter der Voraussetzung, dass es bewusst und reflektiert eingesetzt wird. Ich würde es den Studierenden abraten, dieses Tool als Lernhilfe zu verwenden und nicht als Ersatz für das eigene Denken. Es ist sehr effektiv, zunächst selbst einen Lösungsansatz zu entwickeln und Cursor anschließend zur Überprüfung oder Erklärung zu nutzen. So bleibt der Lerneffekt erhalten.

Außerdem empfiehlt es sich, die Antworten des KI-Assistenten kritisch zu hinterfragen und sich in erster Linie an das Wissen aus den Vorlesungen und Übungen zu orientieren. Um bessere Antworten von der KI zu bekommen, sollte man das Problem möglichst präzise beschreiben und unbedingt prüfen, ob es auch wirklich ein guter Lösungsvorschlag ist, der die Anforderungen der Aufgabe erfüllt.