AI and Databases

COSC 304 – Introduction to Database Systems

Why Use AI when Developing with Databases?

Faster SQL development

- Autogenerates complex queries, joins, subqueries
- Reduces time spent debugging syntax and logic errors

Improved code quality

- Suggests optimized query patterns
- Helps eliminate common pitfalls (repetitive queries, inefficient scans, missing indexes)

Better database design

- Assists in normalizing schemas and identifying design flaws
- Automatically generates ER diagrams and schema documentation

Why Use AI when Developing with Databases? (2)

Performance tuning with guidance

- Interprets execution plans and recommends indexes or rewrites

Rapid prototyping and experimentation

- Generates sample data, schema variations, and test cases
- Accelerates iterative development

Lower barrier for beginners

- Converts natural language to SQL
- Makes databases more accessible for non-experts
- Provides explanations for SQL concepts, relational algebra, normalization, and execution plans
- Helps students practice by generating examples and exercises

Use Case: Question to SQL

One of the most common uses for AI is to help with writing SQL.

Text -to-SQL: Converting a natural language query into SQL

General models like ChatGPT and Claude are very good.

Remember to provide the schema as part of the prompt.

Research challenges: schema linking, ambiguity resolution, complex SQL (joins, subqueries, GROUP BY), robustness

Use Case: AI for Data Cleaning & Preparation

It is often a time-consuming task to write code or manually clean and verify data before it is loaded into a database.

AI can help with:

- Generating scripts to clean the data
- Detecting errors and inconsistencies
- Suggesting transformations
- Generating synthetic datasets for testing
- Identifying outliers or duplicates

AI Inside Database Systems

Many database systems have built-in machine learning and AI support:

- Vector indexes (HNSW, IVF, etc.)
- Embedding-based search
- In-database ML (e.g., BigQuery ML, PostgreSQL extensions)
- Automatic query tuning with ML (e.g., Oracle, SQL Server)

Limitations and Risks

AI mistakes include:

- Hallucinated tables or attributes
- Incorrect join relationships
- Overly complicated queries
- Incorrect assumptions about schema constraints

Data Privacy and Security:

- Understand what data you should not send to AI, including Personally Identifiable Information (PII) and confidential datasets
- Know safe patterns when using AI at school/work

Research and the Future of AI and Databases

Natural Language Interfaces to Databases:

- Text-to-SQL with high accuracy so query by having a conversation
- Improve database design using natural language

Autonomous Database Systems:

- Self-tuning indexes, auto-scaling, predictive performance models

RAG (Retrieval-Augmented Generation) + Databases:

- Improve the performance of how database content is retrieved and provided for AI, and better integration of structured data in improving AI accuracy

Impact of AI on Software Development Jobs

AI is transforming how software is developed. Productivity rises as do expectations for broader skills (prompting, verification, design, AI use).

Benefits:

- **Higher Productivity:** AI-assisted coding accelerates feature development.
- **Improved Code Quality:** Automated review, bug detection, and test generation catch issues earlier.
- **Faster Prototyping:** LLMs generate quick proofs-of-concept and UI mockups.
- **Enhanced Learning:** Developers upskill faster with conversational debugging and explanations.
- **Automation of Repetitive Work:** Documentation, test writing, refactoring, and migrations become easier.
- **More Innovation Time:** Developers focus more on architecture, creativity, and complex problem-solving.

Impact of AI on Software Development Jobs (2)

Challenges:

- **Skill Shift Required:** Developers must learn prompt engineering, model understanding, and AI toolchains.
- **Risk of Over-Reliance:** AI-generated code can introduce security or logic errors.
- **Quality Assurance Needs Increase:** More time spent reviewing AI-generated code for correctness, licensing, and compliance.
- **Security & Privacy Risks:** AI tools may inadvertently leak proprietary code or generate insecure patterns.
- **Job Polarization:** Entry-level tasks are automated, potentially reducing "junior developer" roles while increasing demand for experienced oversight.
- **Ethical and Legal Concerns:** Copyright, bias, and accountability issues require new governance practices.

Developers who use AI effectively will replace those who do not.

Will AI Replace Software Developers?

AI is reducing demand for some types of developer work, including simple CRUD apps, bug fixes, web development and interfaces.

Not yet replacing full-stack software engineering roles as AI:

- Does not understand requirements or business logic: Is the generate code correct, secure, cost-efficient, aligned with user needs?
- Reliably perform systems integration, architecture, and debugging
- Requires generated code to be reviewed and tested including validating correctness, design, and handling of edge cases

AI changes the shape of the developer job market, but does not eliminate developers.

Survey Question: AI and Developers

Question: Do you believe AI will ***replace*** software development jobs?

A) Strongly Agree

B) Agree

C) Neutral

D) Disagree

E) Strongly Disagree

Survey Question: AI and Development

Question: Do you believe AI will ***change*** software development jobs?

A) Strongly Agree

B) Agree

C) Neutral

D) Disagree

E) Strongly Disagree

Survey Question: AI and Jobs

Question: I have concerns that AI will affect my ability to get a job.

A) Strongly Agree

B) Agree

C) Neutral

D) Disagree

E) Strongly Disagree

Survey Question: AI in the Course

Question: I have used AI productively in this course.

A) Strongly Agree

B) Agree

C) Neutral

D) Disagree

E) Strongly Disagree

Conclusion

AI accelerates SQL development, database design, prototyping, and performance tuning.

Natural language interfaces (Text-to-SQL) make databases more accessible to beginners and non-experts.

AI enhances data cleaning, schema generation, and query optimization but can hallucinate tables or produce incorrect joins.

Databases increasingly integrate AI directly (vector indexes, in-database ML, automatic tuning).

Developers must balance productivity gains with risks: privacy, security, correctness, and over-reliance on automated code.

AI is transforming software development roles by shifting emphasis toward architecture, verification, and oversight.

Objectives

- Explain how AI tools support database development (SQL generation, schema design, optimization).
- Describe key concepts behind Text-to-SQL and why schema linking and ambiguity resolution are challenging.
- Identify how AI can assist with data cleaning, transformation, and generating test datasets.
- Recognize common limitations and risks of using AI when working with databases (hallucinations, incorrect joins, privacy/PII concerns).
- Describe how modern databases integrate AI features such as vector indexes, in-database ML, and automatic tuning.
- Discuss how AI is changing software development roles, including the benefits and challenges for developers.


UBC
THE UNIVERSITY OF BRITISH COLUMBIA