

# Inquiry-based Learning of Database Concepts

Wayne Brookes

Faculty of Engineering and Information Technology

University of Technology Sydney, Australia

Email: Wayne.Brookes@uts.edu.au

**Abstract**—In many degree programs, relational database concepts and skills are taught through a combination of lectures combined with tutorials or laboratory sessions, although flipped learning approaches have recently been gaining increasing popularity. This paper describes a different approach using inquiry-based learning to engage students with real, unstructured data-driven challenges. We report on the effectiveness of the inquiry-based learning approach in this context and reflect on challenges for both instructors and students.

**Index Terms**—curriculum design, inquiry-based learning

## I. INTRODUCTION

Inquiry-based learning is an educational approach where students engage in open-ended inquiry while working on complex problems or challenges [4]. The challenge posed to students must be sufficiently open-ended that a variety of responses are possible, and students are given the freedom to choose the direction of inquiry, typically without a specific goal in mind. Students are given some basic knowledge to get started, but later are required to identify their own learning needs and seek out resources to fulfill those needs.

Inquiry-based learning encourages curiosity in students and gives them a sense of agency over their learning. Rather than following a strict content-driven curriculum, students engage in a process of discovery, and identify what concepts they need to learn, and learn them in a sequence that makes sense for them and the challenge they are working with. Students make their own connections between what they are learning, encouraging deeper learning. Lecturers and tutors are no longer focused on delivering content, but rather on guiding students in their learning and helping them to find appropriate learning resources.

However, this creates additional challenges for teachers in designing inquiry-based learning experiences. Although the learning is loosely structured, there are still usually core curriculum concepts that students need to learn. Teachers need to carefully design and guide students' inquiry so that students are exposed to an appropriate range of concepts, and that they are encouraged to go deeper into each concept. It also requires a different kind of student support. Different teams of students may encounter concepts at different times, requiring teachers to be more agile in the way they provide support.

In this paper we report on a trial of using inquiry-based learning for database concepts. We first describe the context and

approach taken, and then consider the preliminary observations from the trial. Finally, we consider some of the challenges in providing this style of education for both educators and students.

## II. CONTEXT

The trial described in this paper took place within a subject in a technology innovation degree program at the University of Technology Sydney. One unique aspect of this degree program is that almost all of the learning is structured as inquiry-based, that is, students spend their time working on projects and challenges of various sizes in all subjects within the degree. There are no traditional lectures, nor are there end of semester examinations.

The particular subject described here focused on introducing students to technologies that allow them to work with data. This included technologies to assist with data collection, storage, cleansing, transformation, querying, analysis and visualization. Students worked in teams and loosely followed a process based around knowledge discovery in databases (KDD) to transform data into knowledge [3]. The subject also had the focus of helping students to see connections between different technology tools, and indeed to build their own connections or networks of tools, which is supported by the KDD process.

While the subject introduced students to a wide range of tools and techniques, this paper focuses on relational database concepts. This differs from similar approaches reported in the literature which focused on using inquiry-based or problem-based learning for data analytics [7] and data-intensive systems [9], however there were similarities in experience and findings.

In the trial reported in this paper, topics that were to be learned by students included the basics of the relational database model (relations, tuples, attributes), including primary keys and foreign keys. Students also learned about database design and modeling through reading and creating simple Entity-Relationship Diagrams (ERDs). Students learned to create SQL queries ranging from basic queries through to table joins and subqueries. In essence, although this wasn't a subject that focused on relational databases in the traditional way, the desired learning outcomes for students had many similarities with a typical database subject.

### III. APPROACH

#### A. The setup

Students were given access to a relatively large extract of a relational database provided by an industry partner that was acting as a client for the student teams. The data principally included information about job seekers and job advertisements. The data set was large — several of the provided tables had in excess of 500,000 rows, and over 30 columns, and there were approximately 15 tables in total. This was key to the inquiry-based approach adopted, as the size and scope of the data needed to be sufficiently large and complex (spread across multiple related tables) that students could not fall back on simpler ways of trying to find insights in data, such as using Excel spreadsheets to sort, filter, summarize or graph the data. With exponentially increasing data volumes, manual data analysis becomes increasingly more impractical [3].

It is essential in inquiry-based learning approaches for students to be engaged with a complex challenge. This is contrary to direct instruction approaches to education, where students learn smaller individual concepts first, before attempting to combine them to solve larger problems.

In the second week of the subject, the students were provided with a database dump from PostgreSQL (the RDBMS used by the client), after they had signed non-disclosure agreements for working with confidential company data. Already this created learning opportunities for students, who needed to learn the technical aspects of how to load the PostgreSQL RDBMS onto their laptops and import the database dump, as well as learning about data privacy and ethics in relation to working with clients. Instructions and in-class support were provided to help students with technical difficulties they faced, given that this was one of the first tasks in the subject, and there was a facilitated class discussion on data ethics and privacy.

Another key aspect of the approach was that students knew the data was real — it was an extract from the client’s production database system. This helped motivate students by knowing that they weren’t working on a fictitious example from a textbook that was contrived with predefined learning moments. Rather they were working with production data that was messy and incomplete, and required them to think critically about the data they were working with. This is in contrast with traditional university-level database subjects, where the lecturer usually provides “clean” data to the students when they are novices. The benefits of having learners work with “unsanitized” data is even recognized at the elementary school level [8].

#### B. The activity

Students were then challenged to work in teams and use the provided database to identify new ways that the industry partner could deliver value for their stakeholders (their clients). Before they began, students received a briefing from a senior

executive of the company and had the opportunity to ask questions about the company’s history, current services and future directions. However the teams were not set a specific task to perform or specific outcome to achieve, other than to identify new services or new ways of providing value that leveraged the data already collected and stored by the client. In many ways, this follows the scenario-based inquiry approach described by Kerven, et al [5].

Teams went through an iterative process to achieve this outcome. While there was some variation between different teams, broadly the process they followed is summarized in Figure 1 on the following page. These stages are briefly described below.

1) *Unstructured inquiry (of data)*: In the beginning, the students had no knowledge of the database structure or contents. Therefore the first phase of work required students to explore the data and the relationships between tables. This necessitated students learning:

- Relational database principles (relations, tuples, attributes, primary keys)
- Relationships (how tables are related to each other, the role of foreign keys)
- Basic SQL queries of the form:

```
SELECT col1, col2
FROM table
WHERE col2 > 10
ORDER BY col1
LIMIT 1000
```

2) *Identify candidate area(s) of focus*: During their open-ended exploration, student teams began to build their own catalog of what the database contained. This helped them to start thinking of possible areas they might focus on, for example, to look at demographic characteristics (gender, age, etc), geographical characteristics (locations of jobs versus locations of job seekers), or job characteristics (jobs or skills in high demand). This provided an opportunity for students to learn:

- Entity-Relationship Diagrams (ERDs) and other kinds of modeling diagrams. Because of the complexity of the database and relationships between tables, students could see intrinsic value in having a standardized way to document the database model to share within their team, and for when communicating with the client. So learning how to create an ERD was something that directly helped them in their project, rather than being a theoretical exercise assigned by a lecturer.

3) *Targeted inquiry (of data)*: Once the teams had identified one or more candidate areas of focus, they then needed to do further extraction and manipulation of data in their chosen area(s). During this phase, teams learned:

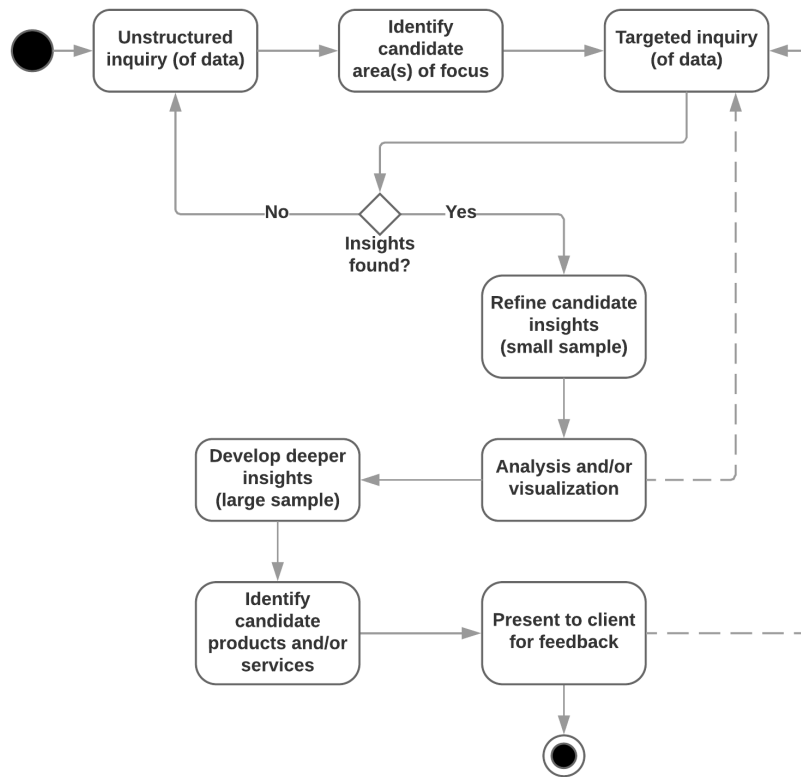


Figure 1. Indicative workflow of inquiry-based approach

- More sophisticated SQL queries, including table joins (inner joins, outer joins) and subqueries. Students quickly realized that their basic single-table SQL queries weren't adequate to find patterns or trends in the data, therefore they were motivated to learn more sophisticated SQL (intermediate level).

4) *Refine candidate insights (small sample) OR reach a dead end*: At this stage, teams had developed some hypotheses about what kinds of insights they might find in the data, and had developed the technical capabilities to be able to test those hypotheses by writing intermediate level SQL queries. There were one of two outcomes here — either the data seemed to support their hypothesis or it didn't. If the data appeared to support their hypothesis (using a small sample), they would continue down the same path to verify their insights on the whole data set. If they reached a dead end, they would either revisit another area of focus already identified, or go back to run new queries to find additional foci (usually the latter).

5) *Analysis and/or visualization*: Once happy with their candidate insights, teams proceeded to do further analysis and/or visualization of the data. Around this phase of the project, students also attended a workshop on Tableau, so many of them created Tableau visualizations using extracts of the data

from the database that helped them to verify whether their hypotheses still held on the whole database. Tableau has the ability to draw data directly from a DBMS, so students were also learning about connecting different tools together to achieve their aims. In the case where the analysis and visualization didn't confirm their hypotheses, students would iterate by going back to an earlier stage of the process, often to perform more targeted inquiry.

6) *Developer deeper insights (large sample)*: Often the analysis or visualization steps allowed students to go further than merely confirming their previous hypotheses, but allowed them develop new or deeper insights, and refine their ideas of what product or service they might be able to suggest to the client that uses the data.

7) *Identify candidate products and/or services*: By this stage, the students had quite a good understanding of the data, and were confident and capable of querying, analyzing and visualizing it. This allowed them to propose one or more new digital products or services that the client might be interested in taking further in their business, and to show the client new ways of providing value to their stakeholders using data they already owned.

8) *Present to client for feedback*: While this is listed as the final step, students actually had several interactions with the

client throughout the journey as it was an iterative process. Students presented their ideas to the client for feedback, and upon receiving the feedback, looped back in the process to further iterate until they were happy with their final proposal.

### C. Summary of key features of approach

This fulfilled the key requirements for inquiry-based learning: that the challenge was open-ended (no specific goal in mind), and that the context was complex and realistic enough to engage students and intrinsically motivate them to want to learn about database concepts to complete the assigned task.

The key features of the approach taken were:

- 1) *Size of data* — too large to use simplistic tools;
- 2) *Complexity of data* — too complex to use simplistic tools (e.g. multiple tables);
- 3) *Using real data* — motivated students by knowing they had to deliver ideas back to a client;
- 4) *Open-ended challenge* — no-one (not even the teaching staff) knew at the beginning what would be achieved;
- 5) *Learning driven by need* — students learned concepts when they needed them and saw direct value, rather than “front loading” students with knowledge;
- 6) *Teaching staff working alongside students* — the learning was a partnership between staff and students, with staff guiding and helping, rather than instructing.

To elaborate on the final point, as part of it being a realistic challenge, even the teaching staff had limited knowledge of the database contents in advance. So students and staff worked together on understanding the data, with staff offering suggestions and guidance, but at no point did staff “have the correct answer” (there was no correct answer). In a more traditional classroom setting, the relationship between teacher and students might be considered as the teacher knowing the answers, and students working towards finding them. In this case, the relationship between teacher and students was one of more genuine cooperation or partnership, working together to tackle the challenge.

Rather than lectures, students were instead provided with access to readings and Open Educational Resources (OERs) such as videos and online course modules to help them understand the basics of database principles and SQL, supplemented with a small amount of in-class instruction and discussion.

## IV. OBSERVATIONS AND COMPARISON

In this section we offer preliminary observations of the inquiry-based approach on the themes of student engagement, student learning, and a comparison with traditional approaches to teaching database concepts. A formal evaluation of this approach is yet to be conducted.

### A. Student engagement

Student engagement was very high with the inquiry-based approach. Students were initially focused on the end goal of proposing a new data-driven product or service, but because the challenge was so complex, they were motivated to learn the underlying principles needed to reach the goal. This was their intrinsic motivation for engaging with the learning materials.

The other key driver of student engagement and motivation was the connection with the client. Students knew they had to be ready to present their findings and ideas to the client every few weeks, providing extrinsic motivation.

The one aspect of engagement which needed some attention was ensuring that all students in the team were equally motivated, and it wasn't just one person doing the work on behalf of the team. This scenario is not unique to inquiry-based learning, but rather a facet of student teamwork, and the approach taken in the trial is discussed further in the section below on challenges for instructors. However, the other side of the story is that Chow and Law [2] found that in school students, student motivation for learning increased by working in teams, in part due to the social learning factor of working with and learning from others, so teamwork can foster both good and bad learning behaviors.

### B. Student learning

By the end of the project, it was evident that all students had a good basic understanding of relational databases, entity-relationship modeling, and SQL. However, due to the nature of the project and assessment, it is not possible to say that students had as thorough a coverage of database topics as if they had been enrolled in a specialist subject on databases. For example, because they were primarily working with an existing database, they learned only a little about designing a new database from scratch, and normalization.

However, while their knowledge may not have been as thorough as if they had studied a specialist database subject, what they did gain was the ability to find information for themselves, and learn new concepts independently. Therefore, in the workforce if they needed more advanced knowledge, they would have the ability to engage in self-learning.

This is often the trade-off with inquiry-based learning. Instructors may feel that students don't learn as much content, however what they gain instead is the ability to learn by themselves when needed. In the current age when there is so much online content available, the latter is a valuable life skill. However there is a balance needed between having basic knowledge and being able to find more advanced knowledge on demand.

### *C. Comparing inquiry-based learning with direct instruction*

Some comparisons have already been made above — with inquiry-based learning, students tend to be more engaged learners than with direct instruction approaches, however they may not learn as much content knowledge, depending upon the project.

In addition to those observations, other points of comparison include:

- With direct instruction approaches, the end goal is usually known in advance. You have a clear idea of what students will learn and what they will be able to do at the end. However because inquiry-based learning is open-ended, even the instructors are not certain where students will end up.
- With direct instruction, students usually work on smaller sub-problems first, and then work up to bigger challenges. With inquiry-based learning, students are given the big challenge first, and then need to break it down to work on smaller sub-problems.
- In an inquiry setting, the order in which students learn concepts is determined more by the needs of the project than by the instructor. However the instructor has a role to play in helping students to break the larger problem down into manageable units and to work through these in a logical sequence so that students don't feel lost and unable to make progress.

## V. CHALLENGES

### *A. Challenges for instructors*

Teaching an inquiry-based subject poses quite different challenges than a traditional subject. Instructors must have good “big picture” knowledge of the concepts being learned, and able to guide students through those concepts in different orders. It is not sufficient to rely on a textbook and follow the chapters in sequence. Thus, the first challenge is that teaching staff must be agile, and able to respond to students' needs when they are at different stages.

The second challenge is around keeping track of students progressing at different paces. Because students are working in teams and each team is taking their own approach, teaching staff need to be mindful that different teams and individuals within those teams will progress at different paces as well. So while it is helpful to bring the whole class together for discussions of key topics, the timing of these is important so that teams who are at different stages can still benefit. This is quite different to a traditional subject where the lecturer dictates the pace as well as the sequence of topics, and it is the students' task to keep up.

The third challenge encountered was around assessment design. With students working in teams (whether in an inquiry-

based setting or not), it can easily happen that one or two students do the majority of technical learning and work, and others choose the role of project manager, or documentation writer to avoid having to go into technical details. In this trial, alongside the major project, students were given separate smaller assessment tasks to demonstrate their own individual learning. For example, in one task, students were asked to take a public data set from the Internet, load it into PostgreSQL and run a range of queries on it that demonstrated their grasp of SQL.

This approach led to an interesting situation where although the students engaged with learning database concepts primarily to meet the needs of the project, their technical competence was in fact being assessed through other tasks alongside the project. This is quite different to the “drill-and-practice” approach sometimes used in teaching languages like SQL. It also gave the opportunity to see how students could apply their learning to new situations. But overall, there is a need for careful assessment design to ensure that individual students are learning.

### *B. Challenges for students*

The biggest challenge for students in this trial was knowing where to begin. When presented initially with a vague problem statement, and a huge database, it was somewhat overwhelming for students who had never touched a DBMS before. This is where the role of the instructor is crucial in offering guidance, particularly in the early stages. For example, even installing the PostgreSQL RDBMS software on students' laptops turned out to be more challenging than anticipated, with each student encountering different problems. However it was an opportunity to encourage peer learning (those who had installed it successfully helping those who were struggling), and even for some students to learn command-line interfaces for the first time (Unix/Mac Terminal and Windows Command Prompt).

To also help with getting started, some class time was devoted to mini-lectures (10-15 minutes maximum) and discussions around basic database concepts and SQL syntax. While students had access to OERs to help them learn this in their own time, being able to work through it in class and discuss it was important, as those who may not have understood the online materials had a chance to ask questions in a supportive environment.

The second main challenge for students was the open-ended and iterative nature of the project work. In a traditional subject, the instructor might assign different topics or themes for teams to explore, choosing ones where the instructor knows there is lots for the students to find. In the inquiry-based approach trialled here, because the instructors did not know much more about the database contents than the students did, it was a genuine journey of exploration, which naturally resulted in some dead ends of inquiry. However, while students may have

found this challenging, it was probably advantageous for their learning, as they had to revisit different tasks multiple times to iterate through new ideas, requiring them to further build their database knowledge and skills.

## VI. CONCLUSION

This paper examined an initial trial of inquiry-based learning of databases, focusing on fundamental database concepts including the relational model, entity-relationship modeling and SQL. While inquiry-based learning is not new, it is still not common in universities as an approach for teaching technical content.

We presented an indicative workflow for the inquiry-based approach used based on observation of the students, and a set of key features that we believe were behind the successful implementation of the approach. These features were the size of the data supplied; the complexity of the data; the use of real client data; the open-ended nature of the challenge; allow students to learn concepts as driven by project needs; and having teaching staff working alongside students.

Further work would be to explore whether other subjects in the IT curriculum are as amenable to inquiry-based learning. For example, some early work by Abernathy, Gabbert and Treu showed promise in using inquiry-based learning in a course on Systems Analysis and on teaching algorithms, but mixed results with Discrete Mathematical Structures and Introduction to Computer Science [1]. Lockwood and Esselstein found success in applying inquiry-based learning in an inverted classroom context for teaching linear algebra [6].

The challenge is in finding problems that are big enough overall and open-ended, yet able to be subdivided into more discrete learning opportunities.

## REFERENCES

- [1] Ken Abernethy, Paula Gabbert, Kevin Treu, Ken Abernethy, Paula Gabbert, and Kevin Treu. Inquiry-based computer science instruction. In *Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education Changing the delivery of computer science education - ITiCSE '98*, volume 30, pages 14–17, New York, New York, USA, 1998. ACM Press.
- [2] Angela Chow and Nancy Law. Measuring motivation in collaborative inquiry-based learning contexts. In *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: The Next 10 Years!*, CSCL '05, pages 68–75. International Society of the Learning Sciences, 2005.
- [3] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The KDD process for extracting useful knowledge from volumes of data. *Commun. ACM*, 39(11):27–34, November 1996.
- [4] Peter Kahn and Karen O'Rourke. Understanding enquiry-based learning. In Terry Barrett, Iain Mac Labhrainn, and Helen Fallon, editors, *Handbook of Enquiry and Problem-based Learning: Irish Case Studies and International Perspectives*, chapter 1. AISHE, Dublin, 2005.
- [5] David Kerven, Kristine Nagel, Stella Smith, Sherly Abraham, and Laura Young. Scenario-Based Inquiry for Engagement in General Education Computing. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE '17*, pages 303–308, New York, New York, USA, 2017. ACM Press.
- [6] Kate Lockwood and Rachel Esselstein. The inverted classroom and the CS curriculum. In *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13*, page 113, New York, New York, USA, 2013. ACM Press.
- [7] Miguel Nunez-del Prado and Rosario Gomez. Learning data analytics through a Problem Based Learning course. In *2017 IEEE World Engineering Education Conference (EDUNINE)*, pages 52–56. IEEE, mar 2017.
- [8] Tia Shelley, Leilah Lyons, Tom Moher, Chandan Dasgupta, Brenda Lopez Silva, Alexandra Silva, Tia Shelley, Leilah Lyons, Tom Moher, Chandan Dasgupta, Brenda Lopez Silva, and Alexandra Silva. Information-building applications. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pages 2123–2132, New York, New York, USA, 2014. ACM Press.
- [9] Tomasz Wiktor Wlodarczyk and Thomas J. Hacker. Problem-Based Learning Approach to a Course in Data Intensive Systems. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pages 942–948. IEEE, dec 2014.