INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 23, No. 2
March 2025
ISSN: 1545-679X

In this issue:

Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is five times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of the ISCAP Conference (https://iscap.us/proceedings) features all papers, abstracts, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the ISCAP conference. All papers, whether award-winners or not, are invited to resubmit for journal consideration after applying feedback from the Conference presentation. Award winning papers are assured of a publication slot; however, all re-submitted papers including award winners are subjected to a second round of three blind peer reviews to improve quality and make final accept/reject decisions. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 35%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to volunteer members of ISCAP who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

Editors

Using LEGO® Brick Data to Teach SQL and Relational Database Concepts

James R. Wolf
jrwolf@ilstu.edu
School of Information Technology
Illinois State University

Abstract

This paper introduces the LEGO® Database, a large natural dataset that can be used to teach Structured Query Language (SQL) and relational database concepts. This dataset is well-suited for introductory and advanced database assignments and end-of-semester group projects. The data is freely available from Kaggle.com and contains eight tables with 633,250 rows of data on 11,673 LEGO® sets sold between 1950 and 2017. As a guiding example, I introduce an example group project assignment designed to provide students hands-on experience with database management and SQL queries. I also discuss tips, suggestions, and lessons learned from using the data for group projects over the past five years. While LEGO® bricks have been widely used in educational settings, including college and computer classrooms, this is the first work to discuss the use of LEGO® data in a college database course.

Keywords: Database Education, Structured Query Language (SQL), Real-World Datasets, Project-based learning, LEGO® for Education, Database Projects

Recommended Citation: Wolf, J., (2025) Using LEGO® Brick Data to Teach SQL and Relational Database Concepts*. Information Systems Education Journal,* 23(2), pp. 29-44.
https://doi.org/10.62273/FYXC7105.

Using LEGO® Brick Data to Teach SQL and Relational Database Concepts

James Wolf

1. INTRODUCTION

There is significant debate among IT educators regarding the best type of database needed to teach Structured Query Language (SQL) and relational database concepts. Some point out that small, simplified databases make it easier for students to grasp fundamental database concepts and SQL syntax without getting overwhelmed (Miao et al., 2019). However, many educators are concerned that "toy" databases may not fully prepare students for the challenges they will face after graduation (Jukic & Gray, 2008; Wagner et al., 2003; Yue, 2013).

Smaller datasets allow students to manually inspect the data and verify the accuracy of their queries (Taipalus et al., 2023). This practice is particularly beneficial for novices as it helps them identify logical errors in their SQL statements and understand the relationship between the query and its output (Miao et al., 2019; Taipalus et al., 2023).

However, these small and simple databases may fail to adequately prepare students for the complexities of real-world database systems, which typically involve large datasets (Yue, 2013). For this reason, many database educators now advocate using large, real-world data in database courses to better prepare students for the complex database systems they will encounter once they enter the workforce (Jukic & Gray, 2008; Wagner et al., 2003; Yue, 2013). As a result, these natural datasets tend to be much larger and more complex than the simplified examples in textbooks. They may provide more realistic learning experiences.

This work introduces database instructors to the Kaggle.com LEGO® database and demonstrates how it can be used to teach database concepts (especially SQL skills) in a college classroom. The data are appropriate for all phases of database instruction but especially well-suited for a semester-long group project. As a guiding example, I introduce a group project assignment designed to provide students with hands-on experience in database management. The detailed project helps students develop SQL and database management technical skills as well as soft skills, such as teamwork and problem-solving.

I have successfully used variations of the detailed LEGO® Database Project in graduate and undergraduate database courses. In addition to describing the LEGO® Database and an example group project assignment, I also discuss teaching tips, suggestions, and lessons learned from using the LEGO® data for group projects over the past five years.

2. LITERATURE REVIEW

Database management and SQL are among IT professionals' most important and sought-after skills. While NoSQL databases have increased in usage, relational databases remain more widely used, and database management and SQL skills continue to be in high demand for IT professionals and a growing number of fields that employ artificial intelligence (AI), business analytics, and data analysis. Recent studies on the required skills for IT professionals show that relational database skills remain in high demand. One of the most important of these skills is the ability to query relational databases using SQL. (Cummings & Janicki, 2021, 2020; Gurcan & Sevik, 2019; Halwani et al., 2022; Li et al., 2021; Radovilsky et al., 2018; Yin & Zhang, 2023)

Database skills are included in approved computing curricula and mandated by computing accrediting bodies. Both ABET's Computing Accreditation Commission and the ACM-AIS IS2020 Task Force model curriculum emphasize the importance of database management in Information Systems degree programs (ABET, 2023; Leidig & Salmela, 2022). The IS2020 report, produced by the Joint Task Force of the Association for Computing Machinery (ACM) and the Association for Information Systems (AIS), is the latest in a series of model curriculum recommendations and guidelines for undergraduate degrees in Information Systems (IS). IS2020 lists the ability to query a relational database as a required competency for Information Systems graduates, specifying that graduates should be able to "*translate user* *stories to SQL statements using (SELECT, FROM, WHERE, ORDER BY, DISTINCT, LIKE, BETWEEN, IN, JOIN, GROUP BY, HAVING, sub-queries, ANY, ALL, UNION)* (Leidig & Salmela, 2022)."

Database Education Challenges
Several studies have found that students find computing boring (Bellino et al., 2021; Biggers et al., 2008; Giannakos et al., 2017; Yardi & Bruckman, 2007; Zaharias, 2009) and unconnected to the world outside of the classroom (Anderson et al., 2008; Bellino et al., 2021). Students often feel that the exercises and assignments from their computing classes are irrelevant to situations encountered in their daily lives (Bellino et al., 2021).

LEGO® Bricks in the College Classroom
Educators from all disciplines have attempted to make learning fun and more hands-on by using LEGO® Serious Play (LSP) exercises in their classes. LEGO® bricks have been employed in a wide array of college classrooms, most often in STEM courses, but also in business and arts classrooms (Benesova, 2023; Geithner & Menzel, 2016; Jensen et al., 2018; Martin-Cruz et al., 2022; Warburton et al., 2022; Wengel, 2020).

LEGO® Bricks in College Computer Classes
LEGO® brick activities have been widely used in computer education. For example, Kurkovsky (2018) highlighted the use of LEGO® bricks in teaching software interface design. Zhang (2016) detailed 12 years of employing LEGO® Robotics to introduce Artificial Intelligence. Lindh and Holgersson (2007) examined the impact of LEGO® Serious Play (LSP) on students' problem-solving in mathematics and technology. Steghöfer et al. (2017) described LEGO®-based workshops for teaching the agile software engineering process and scrum. Morales-Trujillo (2021) described KUALI-Brick as a LEGO® activity for teaching software quality assurance.

Similarly, Kurkovsky (2016) introduced the use of LEGO® bricks to teach test-driven development. Kurkovsky (2015) explored the use of LEGO® Serious Play for teaching software engineering. Fronza et al. (2022) reported a remote coding camp for high school students, adapting LEGO® activities for online engagement. Walsman et al. (2022) employed LEGO® bricks in a virtual learning environment for structural understanding.

Using LEGO® bricks in the computer classroom can create a playful and imaginative atmosphere that many students enjoy. For example, students in Kurkovsky et al. (2019) reported that the LEGO® activities allowed them to understand software development from a different perspective and helped them to visualize and further develop their ideas.

Similarly, student feedback from Kurkovsky (2015) suggested that LEGO® helped improve teamwork and oral communication. Students indicated LEGO increased motivation, promoted creativity, and improved information retention. Many students enjoyed LEGO and looked forward to using them more.

While LEGO® bricks have been widely used in computing education, this is the first paper to discuss the use of LEGO® brick data in database education.

Relevance, not fun, changes perspectives
Fun activities, like those that utilize LEGO® Serious Play, may not be enough to change student attitudes toward computing. Bellino et al. (2021) note that most "fun" interventions are not very useful and do not have a lasting impact on student perceptions. Bellino et al. (2021) noted that students enjoy "fun" interventions but that these interventions did not change students' perceptions of computing as boring/fun. However, Bellino et al. (2021) did find that relevant interventions changed student perceptions of computing as boring/fun.

Students want their studies to feel relevant to their lives, careers, and the world outside the classroom (Bellino et al., 2021). Several studies have found that students do not find computing classes relevant to the real world (Barker et al., 2009; Bellino et al., 2021; Biggers et al., 2008; Kafura & Tatar, 2011; Kapoor & Gardner-McCune, 2018; Yardi & Bruckman, 2007).

Real-World Problems
One of the most effective ways to make learning relevant and connect student learning with the outside world is by having students work on real-world problems (Hsu et al., 2018). Real-world problems can help students develop their problem-solving skills, learn to think critically and develop their ability to collaborate with others. Additionally, real-world problems can help students see the relevance of computational thinking in their own lives and future careers (Hsu et al., 2018). Similarly, using a real-world dataset with a well-known domain (e.g., LEGO® bricks) could provide a connection to everyday life that students feel is often missing from computing classes (Bellino et al., 2021; Jukic & Gray, 2008).

Education research shows that a problem-solving curriculum based on real-world settings can lead to better intellectual curiosity and attitudes toward education (Bellino et al., 2021). Students are more motivated to learn when the subject matter is relevant to their personal lives (Ormrod & Davis, 2004) and learn more when the material is interesting (Ormrod & Davis, 2004).

The Declarative Nature of SQL

Novice SQL programmers have difficulty with SQL queries that require GROUP BY with and without HAVING, NATURAL JOINS, simple subqueries, correlated subqueries, and self-joins (Ahadi et al., 2015, 2016; Miedema et al., 2023; Migler & Dekhtyar, 2020; Taipalus et al., 2018).

Sadiq et al. (2004) suggested that these difficulties stem from the declarative nature of SQL. SQL requires that students think in terms of sets rather than step-by-step procedures. Echoing this, Celko (2008) believes that a procedural programming mindset keeps SQL novices from taking full advantage of the power of SQL and other declarative languages. This mindset poses a hurdle for many beginners. Ahadi et al. (2015) suggest that novices might make errors due to a procedural approach to constructing queries rather than embracing the set-based logic of SQL.

To affect the needed change in mindset, novice SQL programmers need both instruction and opportunities for practice. The LEGO® Database combines the fun of LEGO® bricks and the relevance of a large database of real-world data with a well-known domain.

The LEGO® Group was founded in Billund, Denmark, in 1932 by Ole Kirk Kristiansen and is now one of the world's largest manufacturers of toys (LEGO.com, 2024a). The LEGO® name derives from the Danish words *Leg* and *Godt*, which means "Play Well" (LEGO.Com, 2024a). The company is best known for its LEGO® bricks. LEGO® bricks are small, interlocking plastic blocks in various shapes, sizes, and colors. LEGO® bricks can be connected to create countless models and structures. LEGO® bricks are typically sold in sets that allow builders to build a specific object (LEGO.Com, 2024a). The most popular sets are a mix of homegrown themes LEGO® Icons, a range for older builders, LEGO® City and LEGO® Technic™, and entertainment IPs like Star Wars™ and Harry Potter™ (LEGO.Com, 2024b).

3. THE LEGO® DATABASE

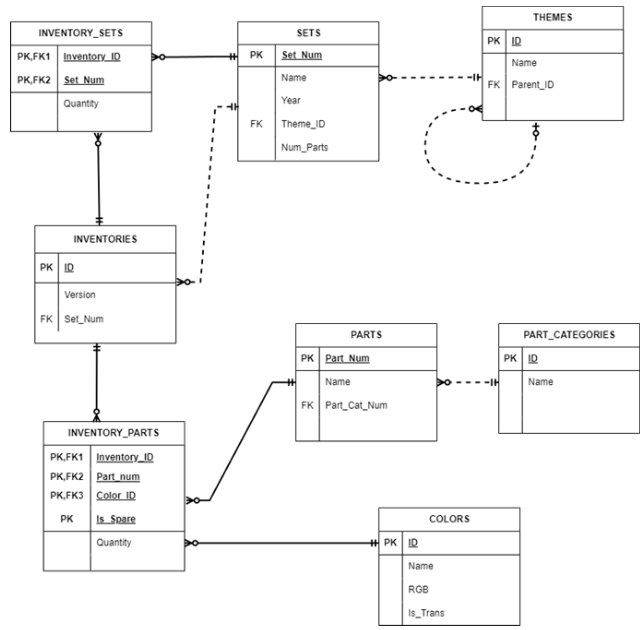

Figure 1: A Basic Entity Relationship Diagram of the LEGO Database

The LEGO® database has eight tables with 633,250 rows of data on 11,673 LEGO® sets sold between 1950 and July 2017. Please see Appendix A for more details.

Table	**Rows**	**Columns**
Sets	11,673	5
Colors	135	4
Themes	614	3
Inventories	11,681	3
Inventory_Parts	580,251	6
Inventory_Sets	2,846	3
Part_Categories	57	2
Parts	25,993	3
Total	**633,250**	**29**

Table 1: List of Tables in the LEGO® database

LEGO bricks have become a global phenomenon, and several online sites cater to LEGO fans. One of the best of these sites is Rebrickable.com (Brick Land, 2022). Rebrickable.com allows owners of LEGO sets to see the other LEGO sets they can build from the sets and parts they already own (https://rebrickable.com/about/). The database contains information on which parts are included in different LEGO® sets (*LEGO Database*, 2017). The data for the Kaggle.com LEGO® Database was obtained from Rebrickable.com and uploaded to Kaggle.com by Rachael Tatman, a language technology educator who previously worked as a developer advocate and data scientist at Kaggle.com (*Tatman*, 2024).

4. AN EXAMPLE ASSIGNMENT

To illustrate the LEGO® database's classroom potential, I present and discuss a hands-on database assignment. Please see Appendix B for the full text of the assignment. The LEGO® Database Project has six parts and seven deliverables, each with unique requirements and point values. This project typically accounts for 10% of the overall course grade.

Section 1. Group Contract: The project begins with team formation and the completion of a Group Contract. This contract delineates the team's roles, responsibilities, and expectations, thus ensuring a unified approach.

The instructor assigned all groups, and no formal roles were assigned. Students determined formal or informal roles after posting the assignment and after a lecture on group work. In addition to a lecture on *Building an Effective Team*, teams were required to prepare and assent to a group contract outlining modes of communication, expectations, and a conflict resolution framework.

Section 2. Data Model: Using the LEGO® dataset from Kaggle, students analyzed the data and created an Entity-Relationship Diagram (ERD) employing the crow's foot notation based on the provided crude ERD. This task aimed to develop skills for understanding data relationships and schema design.

Section 3. Creating and Loading Kaggle Data: Students employ Data Definition Language (DDL) and Data Manipulation Language (DML) to create database tables according to the ERD, load data, and establish key relationships. This phase offers practical experience in database creation and management.

Section 4. Querying the Data: This section challenges students to write SQL queries for data extraction, ranging from basic retrieval to complex queries involving aggregate functions and subqueries. It tests proficiency in data retrieval using SQL commands and functions.

Section 5. Entering New Data: Students manually input data for a specific LEGO® set (e.g., a Pirate Ship) to simulate real-world database updating scenarios. This tests students' understanding of the existing database structure and helps develop integration skills.

Section 6. Feedback: The project concludes with reflective feedback. Students write statements outlining their contributions and evaluating their peers, fostering self-awareness and peer evaluation skills.

Assessment is based on participation, submission quality, and peer feedback and is aimed at promoting engagement and a deeper understanding of database concepts. The methodology emphasizes practical application, collaborative learning, and reflective practice within a structured educational framework, bolstering technical and soft skills, such as teamwork and critical thinking.

5. TEACHING TIPS

Group projects and collaborative assignments are common in collegiate database courses. If done well, group work can enhance student enjoyment, engagement, and learning (Johnson & Johnson, 2008; Murphy et al., 2020). However, if done poorly, group work can add to student frustration, disengagement, and group failure (Thiemann, 2022; Wolf, 2011).

Group Formation

Given the importance of getting groups right, group formation has been widely studied. While some studies report the advantages of homogeneous groups (Müller et al., 2024) or even that group diversity does not impact results (Horwitz & Horwitz, 2007), most research suggests that diverse groups outperform non-diverse groups (Chen et al., 2019; Cheng et al., 2008; Horwitz & Horwitz, 2007; Summers & Volet, 2008; Yang et al., 2022).

It is important to prioritize building teams for success. I typically create teams of three or four students. For a class of 24, I create eight groups of three. Strategically, I distribute the strongest SQL programmers and students who might need more team support. I place the remaining students randomly or based on stated student preferences. Before team formation, I solicit student input on desired teammates.

My experience shows that placing all high performers together or all those who struggle together leads to uneven outcomes. By mixing skill levels, I find that all groups are more likely to complete the project successfully.
To promote problem-solving within teams, I encourage students to ask their teammates for help before contacting me with project difficulties.

I recently implemented a policy where students must copy their teammates on any questions sent via email. This fosters transparency and potentially allows teammates to solve the problems among themselves. Additionally, I often delay before responding to a team member's question, giving their peers a chance to offer solutions and reinforcing a sense of team responsibility.

Encouraging Teamwork

Following Johnson and Johnson's (2008) advice, I have built in interdependence, time for teamwork skills, and individual accountability in the group project. I have created interdependence by providing each student group with group-specific Oracle credentials. All group members have access to the same Oracle account, and all SQL-related group work must be completed using the group account. Also, the assignment focuses on interdependence. The tables must be created before the data can be loaded. The data must be in place before the queries in section 4 can be completed or before the new set's data can be loaded in section five.

I devote time to teamwork in several ways. First, before assigning the project, I lecture on teamwork in class. I also incorporate several strategies for mitigating negative group aspects. Each team must create a group contract that spells out expectations and how to handle group conflicts.

To encourage teamwork and team bonding, I have students complete in-class assignments together and establish team contracts outlining communication expectations, conflict resolution protocols, and project roles. Once the group project is assigned, teams must sit together during class meetings and work collaboratively on in-class assignments and project components.

The project's deliverables are staggered over several weeks, and I check in with each team during each class meeting after the project is assigned.

Finally, I build individual accountability into the assignment by requiring each team to summarize each member's contribution. Each student must complete an individual learning reflection and a post-project survey asking them to rate their group members and themselves on several aspects of group interactions.

Tools and Technologies Used

For this assignment, students used Oracle 19c and SQL developer. These tools were already used throughout both the undergraduate and graduate courses. In addition, the SQL developer has easy-to-use import functionality, which allows the students to import the Kaggle LEGO® data directly into Oracle. Throughout the courses, students use individual Oracle accounts. All group members were given access to a group account on the university server for the group assignment. Another benefit of using SQL Developer is that it allows for command echoing via the SET ECHO utility. This allows students to submit a single plain text file showing both SQL and query results.

Assessment Criteria

Learning outcomes and assignment grades were assessed through direct and indirect measures. The direct measures included the evaluation of the technical correctness of all deliverables—the ERD, SQL scripts, and reflective feedback. The indirect measures involved self-assessments and peer evaluations, which were measured using online surveys.

Technical Correctness

Errors in SQL queries can be classified into four error categories: syntax errors, semantic errors, logical errors, and complications (Ahadi et al., 2015, 2016; Miedema et al., 2023; Migler & Dekhtyar, 2020; Taipalus et al., 2018). *Syntax errors* are errors in the formatting and structure of the SQL code that prevent the database management system from understanding and executing the query (Ahadi et al., 2015, 2016; Taipalus et al., 2018). *Semantic errors* are errors where the SQL code is syntactically correct but does not produce the intended results for any given data demand (Brass & Goldberg, 2006; Taipalus et al., 2018). *Logical errors* are errors where the SQL code is syntactically correct but does not produce the intended results for a particular data demand (Taipalus et al., 2018). C*omplications* are queries that return the correct result table but are unnecessarily complex in their execution (Brass & Goldberg, 2006; Taipalus et al., 2018). Taipalus et al. (2018) use the term *exemplary* to denote queries without errors or complications.

When judging the technical correctness of the SQL coding segments of the assignments, I use the following criteria. Exemplary code receives full credit, and SQL complications receive only minor deductions. Semantic and logical errors, depending on the severity of the error, receive partial credit. SQL with syntax errors usually incurs significant deductions.

Student Engagement

Student engagement is primarily measured via student self-assessments and peer evaluations. Both the self-assessments and peer evaluations were measured using online surveys. The surveys asked about leadership and participation rates in group discussions, consistency in meeting project deadlines, the level of effort, and the portion of the assignment completed.

6. DISCUSSION

There is significant debate among IT educators regarding the best type of database needed to teach Structured Query Language (SQL) and relational database concepts.

Simple Databases

Simple databases are easy for beginning students to understand and visualize (Gudivada et al., 2007). These small databases allow students to focus on concepts rather than data (Gudivada et al., 2007; Yue, 2013) and easily identify SQL mistakes (Miao et al., 2019).

Yue (2013) found that students are more engaged by databases with a readily understood business domain, a relatively simple structure, and a realistic but manageable amount of data. Miao et al. (2019) note that using large datasets to explain SQL errors is often ineffective, especially in educational settings. Instead, they suggest smaller, more focused counterexamples offer a more efficient and understandable way to illustrate the source of the error.

The databases used in most college database classes are small and almost "toy-like" (Yue, 2013). They are intentionally small to aid instruction but have a toy-like feel that adds to the disconnect between classroom exercises and real life. Utilizing the Teradata University Network's DMS to examine databases from several popular textbooks, Yue (2013) found that textbook database tables tended to be small, simple, and lacked advanced features. Confirming Wagner, Shoop, and Carlis (2003), Yue (2013) found that most textbook databases utilized the employee-department-project, student-course-enrollment, or similar domains.

Natural Databases

Natural learning environments are those that more closely resemble real-world work environments, while manufactured learning environments are more controlled and structured (Taipalus & Seppänen, 2020).

Natural learning environments may better prepare students for future work environments and help them develop problem-solving and critical-thinking skills. However, they can be more difficult for students to learn and may not provide all students with the necessary structure and support (Taipalus & Seppänen, 2020).

A Middle Ground

Taipalus and Seppänen (2020) suggest that the best approach to teaching SQL is to use a mix of natural and unnatural learning environments. This allows students to benefit from the advantages of both types of environments and helps them develop the skills they need to be successful in the workplace. One common solution is to employ small databases in the early stages of a database course and then switch to a larger, more complex database for assignments and projects after students have mastered the basics (Seyed-Abbassi et al., 2007; Wagner et al., 2003). Silberschatz et al. (2011) offer another approach, using tables with a few rows for early course examples but increasing the number of rows for more advanced exercises (Seyed-Abbassi et al., 2007; Taipalus et al., 2023; Wagner et al., 2003).

Gudivada et al. (2007) suggest what could be considered a middle ground, using a subset of a large natural database. They describe using a subset of the available product data from Amazon.com: only books within the "Computers & Internet" category for their relational database course. Gudivada et al. (2007) note that datasets used for database instruction should go beyond simple textbook examples, offering students a realistic and challenging experience that mirrors real-world database characteristics and complexities while ensuring that the scale of the project remains manageable within the timeframe of a semester-long course.

A related option, not mentioned in the literature but employed by the author, is using privileges, synonyms, and tailored views for instruction and examples. The views can be altered and increased in size and complexity once students have mastered basic concepts and syntax. This method, like Gudivada et al. (2007), has the added benefit of acquainting students with the tables and data domain before adding complexity and volume.

7. CONCLUSIONS

Educational research suggests that focusing on problem-solving in the real world may foster intellectual curiosity and motivation, attitudes toward schooling, and academic achievement (Angeli et al., 2016; Wolfe & Brandt, 1998). When solving real-world problems, students show greater curiosity, motivation, attitudes toward learning, and greater achievement. Focusing on real-world problems can make computational thinking more relevant and keep students engaged and interested in the subject (Wolfe & Brandt, 1998).

This work introduces the Kaggle.com LEGO® Database and demonstrates how it can be used to teach database concepts (especially SQL skills) in a college classroom. The LEGO® Database is a large, real-world dataset. The dataset is complex enough for advanced student assignments. However, since the domain is familiar to most students, the LEGO® Database is also appropriate for beginning exercises. Assignments using the LEGO® Database have been well received by students.

While relational database education and SQL are the focus of this work, the LEGO® Database is also useful for a wide variety of undergraduate and graduate courses, including statistics, data science, and research methods.

8. ACKNOWLEDGEMENTS

I want to thank Dr. Rachael Tatman for placing the LEGO® Database on Kaggle.com and for the incredibly helpful data descriptions. I would also like to thank the hundreds of students who have worked through the example assignment and provided valuable feedback.

9. REFERENCES

ABET. (2023). *Criteria for accrediting computing programs, 2023 – 2024*. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2023-2024/

Ahadi, A., Prior, J., Behbood, V., & Lister, R. (2015). A quantitative study of the relative difficulty for novices of writing seven different types of SQL queries. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 201–206. https://doi.org/10.1145/2729094.2742620

Ahadi, A., Prior, J., Behbood, V., & Lister, R. (2016). Students' semantic mistakes in writing seven different types of SQL queries. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 272–277. https://doi.org/10.1145/2899415.2899464

Anderson, N., Lankshear, C., Timms, C., & Courtney, L. (2008). 'Because it is boring, irrelevant and I do not like computers': Why high school girls avoid professionally-oriented ICT subjects. *Computers and Education*, *50*(4), 1304–1318. https://doi.org/10.1016/j.compedu.2006.12.003

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47–57.

Barker, L. J., McDowell, C., & Kalahar, K. (2009). Exploring factors that influence computer science introductory course students to persist in the major. *ACM SIGCSE Bulletin*, *41*(1), 153–157. https://doi.org/10.1145/1539024.1508923

Bellino, A., Herskovic, V., Hund, M., & Munoz-Gama, J. (2021). A real-world approach to motivate students on the first class of a computer science course. *ACM Transactions on Computing Education*, *21*(3), 1–23. https://doi.org/10.1145/3445982

Benesova, N. (2023). LEGO® Serious Play® in management education. *Cogent Education*, *10*(2), 2262284. https://doi.org/10.1080/2331186X.2023.2262284

Biggers, M., Brauer, A., & Yilmaz, T. (2008). Student perceptions of computer science: A retention study comparing graduating seniors with CS leavers. *ACM SIGCSE Bulletin*, *40*(1), 402–406. https://doi.org/10.1145/1352322.1352274

Brass, S., & Goldberg, C. (2006). Semantic errors in SQL queries: A quite complete list. *Journal of Systems and Software*, *79*(5), 630–644. https://doi.org/10.1016/j.jss.2005.06.028

Celko, J. (2008). *Joe Celko's thinking in sets: Auxiliary, temporal, and virtual tables in SQL*. Morgan Kaufmann.

Chen, C.-M., & Kuo, C.-H. (2019). An optimized group formation scheme to promote collaborative problem-based learning. *Computers & Education*, *133*, 94–115. https://doi.org/10.1016/j.compedu.2019.01.011

Cheng, W., Lam, S., & Chung-Yan Chan, Joanne. (2008). When high achievers and low achievers work in the same group: The roles of group heterogeneity and processes in project-based learning. *British Journal of Educational Psychology*, 78(2), 205–221. https://doi.org/10.1348/000709907X218160

Cummings, J., & Janicki, T. (2021). Survey of technology and skills in demand: 2020 update. *Journal of Information Systems Education*, *32*(2), 150–159.

Cummings, J., & Janicki, T. N. (2020). What skills do students need? A multi-year study of IT/IS knowledge and skills in demand by employers. *Journal of Information Systems Education*, *31*(3), 208.

Fronza, I., Corral, L., Wang, X., & Pahl, C. (2022). Keeping fun alive: An experience report on running online coding camps. *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, 165–175. https://doi.org/10.1145/3510456.3514153

Geithner, S., & Menzel, D. (2016). Effectiveness of learning through experience and reflection in a project management simulation. *Simulation and Gaming*, *47*(2), 228–256. https://doi.org/10.1177/1046878115624312

Giannakos, M. N., Pappas, I. O., Jaccheri, L., & Sampson, D. G. (2017). Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies*, *22*(5), 2365–2382. https://doi.org/10.1007/s10639-016-9538-1

Gudivada, V. N., Nandigam, J., & Tao, Y. (2007). Enhancing student learning in database courses with large data sets. *2007 37th Annual Frontiers in Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, S2D-13. https://ieeexplore.ieee.org/abstract/document/4418135/

Gurcan, F., & Sevik, S. (2019). Expertise roles and skills required by the software development industry. *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, 1–4. https://ieeexplore.ieee.org/abstract/document/8965571/

Halwani, M. A., Amirkiaee, S. Y., Evangelopoulos, N., & Prybutok, V. (2022). Job qualifications study for data science and big data professions. *Information Technology and People*, *35*(2), 510–525.

Horwitz, S. K., & Horwitz, I. B. (2007). The Effects of Team Diversity on Team Outcomes: A Meta-Analytic Review of Team Demography. *Journal of Management*, *33*(6), 987–1015. https://doi.org/10.1177/0149206307308587

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, *126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Jensen, C. N., Seager, T. P., & Cook-Davis, A. (2018). LEGO® SERIOUS PLAY® In Multidisciplinary Student Teams. *International Journal of Management and Applied Research*, *5*(4), 264–280. https://doi.org/10.18646/2056.54.18-020

Johnson, R. T., & Johnson, D. W. (2008). Active learning: Cooperation in the classroom. *The Annual Report of Educational Psychology in Japan*, *47*, 29–30. https://doi.org/10.5926/arepj1962.47.0_29

Jukic, N., & Gray, P. (2008). Using real data to invigorate student learning. *ACM SIGCSE Bulletin*, *40*(2), 6–10. https://doi.org/10.1145/1383602.1383604

Kafura, D., & Tatar, D. (2011). Initial experience with a computational thinking course for computer science students. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 251–256. https://doi.org/10.1145/1953163.1953242

Kapoor, A., & Gardner-McCune, C. (2018). Considerations for switching: Exploring factors behind CS students' desire to leave a CS major. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 290–295. https://doi.org/10.1145/3197091.3197113

Kurkovsky, S. (2015). Teaching Software Engineering with LEGO Serious Play. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 213–218. https://doi.org/10.1145/2729094.2742604

Kurkovsky, S. (2016). A LEGO-based Approach to Introducing Test-Driven Development. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 246–247. https://doi.org/10.1145/2899415.2925500

Kurkovsky, S. (2018). Using LEGO to teach software interfaces and integration. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 371–372. https://doi.org/10.1145/3197091.3205831

Kurkovsky, S., Ludi, S., & Clark, L. (2019). Active Learning with LEGO for Software Requirements. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 218–224. https://doi.org/10.1145/3287324.3287444

LEGO.Com. (2024a). *The LEGO Group history—About Us—LEGO.com*. Retrieved July 30, 2024, from https://www.lego.com/en-us/aboutus/lego-group/the-lego-group-history

LEGO.Com. (2024b, May 23). *LEGO delivered topline growth and outpaced market in 2023—About Us*. LEGO.Com. Retrieved July 30, 2024 from https://www.lego.com/en-us/aboutus/news/2024/march/lego-delivered-topline-growth-and-outpaced-market-in-2023

LEGO Database. (2017). Retrieved July 30, 2024 from https://www.kaggle.com/datasets/rtatman/lego-database

Leidig, P. M., & Salmela, H. (2022). The ACM/AIS IS2020 Competency Model for Undergraduate Programs in Information Systems: A Joint ACM/AIS Task Force Report. *Communications of the Association for Information Systems*, *50*(1). https://doi.org/10.17705/1CAIS.05021

Li, G., Yuan, C., Kamarthi, S., Moghaddam, M., & Jin, X. (2021). Data science skills and domain knowledge requirements in the manufacturing industry: A gap analysis. *Journal of Manufacturing Systems*, *60*, 692–706. https://doi.org/10.1016/j.jmsy.2021.07.007

Lindh, J., & Holgersson, T. (2007). Does LEGO training stimulate pupils' ability to solve logical problems? *Computers and Education*, *49*(4), 1097–1111. https://doi.org/10.1016/j.compedu.2005.12.008

Martin-Cruz, N., Martin-Gutierrez, A., & Rojo-Revenga, M. (2022). A LEGO® Serious Play activity to help teamwork skills development amongst business students. *International Journal of Research & Method in Education*, *45*(5), 479–494. https://doi.org/10.1080/1743727X.2021.1990881

Miao, Z., Roy, S., & Yang, J. (2019). Explaining Wrong Queries Using Small Examples. *Proceedings of the 2019 International Conference on Management of Data*, 503–520. https://doi.org/10.1145/3299869.3319866

Miedema, D., Fletcher, G., & Aivaloglou, E. (2023). Expert Perspectives on Student Errors in SQL. *ACM Transactions on Computing Education*, *23*(1), 1–28. https://doi.org/10.1145/3551392

Migler, A., & Dekhtyar, A. (2020). Mapping the SQL Learning Process in Introductory Database Courses. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 619–625. https://doi.org/10.1145/3328778.3366869

Morales Trujillo, M. E. (2021). Learning Software Quality Assurance with Bricks. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 11–19. https://doi.org/10.1109/ICSE-SEET52601.2021.00010

Müller, A. M., Röpke, R., Konert, J., & Bellhäuser, H. (2024). Investigating group formation: An experiment on the distribution of extraversion in educational settings. *Acta Psychologica*, *242*, 104111. https://doi.org/10.1016/j.actpsy.2023.104111

Murphy, M. C., Mejia, A. F., Mejia, J., Yan, X., Cheryan, S., Dasgupta, N., Destin, M., Fryberg, S. A., Garcia, J. A., Haines, E. L., Harackiewicz, J. M., Ledgerwood, A., Moss-Racusin, C. A., Park, L. E., Perry, S. P., Ratliff, K. A., Rattan, A., Sanchez, D. T., Savani, K., … Pestilli, F. (2020). Open science, communal culture, and women's participation in the movement to improve science. *Proceedings of the National Academy of Sciences*, *117*(39), 24154–24164. https://doi.org/10.1073/pnas.1921320117

Ormrod, J. E., & Davis, K. M. (2004). *Human learning*. (pp. 1-5). London: Merrill.

Radovilsky, Z., Hegde, V., Acharya, A., & Uma, U. (2018). Skills requirements of business data analytics and data science jobs: A comparative analysis. *Journal of Supply Chain and Operations Management*, *16*(1), 82–101.

Sadiq, S., Orlowska, M., Sadiq, W., & Lin, J. (2004). SQLator: An online SQL learning workbench. *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 223–227. https://doi.org/10.1145/1007996.1008055

Seyed-Abbassi, B., King, R., & Wiseman, E. (2007). The Development of a Teaching Strategy for Implementing a Real-World Business Project into Database Courses. *Journal of Information Systems Education*, *18*(3).

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *Database system concepts* (7th Edition). McGraw-Hill Education.

Steghöfer, J.-P., Burden, H., Alahyari, H., & Haneberg, D. (2017). No silver brick: Opportunities and limitations of teaching Scrum with Lego workshops. *Journal of Systems and Software*, *131*, 230–247. https://orcid.org/0000-0002-1811-0123

Summers, M., & Volet, S. (2008). Students' attitudes towards culturally mixed groups on international campuses: Impact of participation in diverse and non-diverse groups. *Studies in Higher Education*, 33(4), 357–370. https://doi.org/10.1080/03075070802211430

Taipalus, T. (2019). A notation for planning SQL queries. *Journal of Information Systems Education*, *30*(3), 160–166.

Taipalus, T., Miedema, D., & Aivaloglou, E. (2023). Engaging Databases for Data Systems Education. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 334–340. https://doi.org/10.1145/3587102.3588804

Taipalus, T., & Seppänen, V. (2020). SQL Education: A Systematic Mapping Study and Future Research Agenda. *ACM Transactions on Computing Education*, *20*(3), 1–33. https://doi.org/10.1145/3398377

Taipalus, T., Siponen, M., & Vartiainen, T. (2018). Errors and Complications in SQL Query Formulation. *ACM Transactions on Computing Education*, *18*(3), 1–29. https://doi.org/10.1145/3231712

Tatman, R. (2024). Dr. Rachael Tatman. Retrieved July 30, 2024 from https://www.rctatman.com/

TheBrickLand.Com. (2022). 13 LEGO Websites everyone should know about. Brick Land. https://www.thebrickland.com/13-lego-websites-everyone-should-know-about/

Thiemann, P. (2022). The Persistent Effects of Short-Term Peer Groups on Performance: Evidence from a Natural Experiment in Higher Education. *Management Science*, *68*(2), 1131–1148. https://doi.org/10.1287/mnsc.2021.3993

Wagner, P. J., Shoop, E., & Carlis, J. V. (2003). Using scientific data to teach a database systems course. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, 224–228. https://doi.org/10.1145/611892.611975

Walsman, A., Zhang, M., Kotar, K., Desingh, K., Farhadi, A., & Fox, D. (2022). *Break and Make: Interactive Structural Understanding Using LEGO Bricks*. https://doi.org/10.48550/ARXIV.2207.13738

Warburton, T., Brown, J., & Sandars, J. (2022). The use of LEGO® SERIOUS PLAY® within nurse education: A scoping review. *Nurse Education Today*, *118, 105528.* https://doi.org/10.1016/j.nedt.2022.105528

Wengel, Y. (2020). LEGO® Serious Play® in multi-method tourism research. *International Journal of Contemporary Hospitality Management*, *32*(4), 1605–1623.

Wolf, J. (2011). A reexamination of gender-based attitudes toward group projects: Evidence from the Google online marketing challenge. *Computers in Human Behavior*, *27*(2), 784–792. https://doi.org/10.1016/j.chb.2010.11.001

Wolfe, P., & Brandt, R. (1998). What Do We Know from Brain Research?. *Educational Leadership*, *56*(3), 8–13.

Yang, Y., Tian, T. Y., Woodruff, T. K., Jones, B. F., & Uzzi, B. (2022). Gender-diverse teams produce more novel and higher-impact scientific ideas. *Proceedings of the National Academy of Sciences*, 119(36), e2200841119. https://doi.org/10.1073/pnas.2200841119

Yardi, S., & Bruckman, A. (2007). What is computing?: Bridging the gap between teenagers' perceptions and graduate students' experiences. *Proceedings of the Third International Workshop on Computing Education Research*, 39–50. https://doi.org/10.1145/1288580.1288586

Yin, J., & Zhang, W. (2023). Research on Talent Demand Analysis in Big Data Related Fields

Based on Text Mining. *Proceedings of the 2023 6th International Conference on Information Management and Management Science*, 33–40. https://doi.org/10.1145/3625469.3625493

Yue, K.-B. (2013). Using a semi-realistic database to support a database course. *Journal of Information Systems Education*, *24*(4), 327.

Zaharias, P. (2009). Usability in the context of e-learning: A framework augmenting 'traditional' usability constructs with instructional design and motivation to learn. *International Journal of Technology and Human Interaction (IJTHI)*, *5*(4), 37–59.

Zhang, H., Lin, L., Zhan, Y., & Ren, Y. (2016). The Impact of Teaching Presence on Online Engagement Behaviors. *Journal of Educational Computing Research* 54(7), 887–900. https://doi.org/10.1177/0735633116648171

APPENDIX A.

LEGO® Database Table Description

Please note that all file and attribute descriptions are from the Kaggle.com LEGO® Database site: https://www.kaggle.com/datasets/rtatman/lego-database

The LEGO® Database has eight tables, which are:

Sets: This table contains information about the LEGO sets, such as their name, year, number of parts, and theme.

The sets.csv file has 11,673 rows and five columns. Each column represents a different attribute of a LEGO set. Here is a brief description of each column:

- set_num: The unique identification number of the set, consisting of letters and numbers.
- name: The set's name, such as "Fire Truck" or "Batwing Battle Over Gotham City".
- year: The year when the set was released, ranging from 1950 to 2017.
- theme_id: The identification number of the theme the set belongs to, such as "City" or "Batman".
- num_parts: The number of parts the set contains, ranging from 1 to 5922.

Colors: This table contains information about the LEGO colors, such as their name, RGB value, and whether they are transparent.

The file colors.csv has 135 rows and four columns. The columns are:

- id: a unique identifier for each color (integer)
- name: the name of the color (string)
- rgb: the hexadecimal code for the color (string)
- is_trans: a boolean value indicating whether the color is transparent or not (string)

Themes: This table contains information about the LEGO themes, such as their name and parent theme.

- themes.csv: This file contains 614 rows and three columns. It lists the theme names, theme IDs, and parent theme IDs of the LEGO themes.

id: Theme unique ID. (integer)
name: Name of the theme. (string)
parent_id: Unique ID for the larger theme, if there is one. (integer)

Inventories: This table contains information about the inventories of the LEGO sets, such as their set number, version, and number of parts.

The file inventories.csv in the LEGO database has 11,681 rows and three columns. The columns are:

- id: The unique identifier of the inventory (integer)
- version: The version of the inventory (integer)
- set_num: The set number of the

inventory (string)
id: Unique ID for this inventory entry.
version: Version number.
set_num: Set number (form `sets.csv`).

Inventory_Parts: This table contains information about the parts in each inventory, such as their part number, color, quantity, and whether they are spare or not.

The file inventory_parts.csv has 580,251 rows and six columns. The columns are:

- inventory_id: The ID of the inventory the part is in (integer)
- part_num: The ID of the part (string)
- color_id: The ID of the color of the part (integer)
- quantity: The quantity of the part in the inventory (integer)
- is_spare: Whether the part is a spare or not (string)

inventory_id: Unique ID for the inventory this part is appearing in. This is the same as the id value in `inventories.csv.`
part_num: Unique ID for the part, as per `parts.csv.`
color_id: Unique ID for the color, as per `colors.csv.`
quantity: The number of copies of this part included in the set!
is_spare: Whether or not this is a spare part. Spare parts are additional parts not needed to finish the set.

Inventory_Sets: This table contains information about the sets in each inventory, such as their set number, quantity, and whether they are spare or not.

The file inventory_sets.csv has 2,846 rows and three columns. The columns are:
- inventory_id: The ID of the inventory the set belongs to (integer)
- set_num: The set number (string)
- quantity: The quantity of the set in the inventory (integer)

inventory_id: Unique inventory ID from `inventories.csv.`
set_num: Unique set ID from `sets.csv.`
quantity: The quantity of the inventory included.

Part_Categories: This table contains information about the categories of the LEGO parts, such as their name and ID.

The part_categories.csv file has 57 rows and two columns. The columns are:
- id: contains the unique identifier for each part category (integer).
- name: contains the name of each part category (string).

Parts: This table includes information on Lego parts, including a unique ID number, the name of the part, and what part category it is from.

The file parts.csv has 25,993 rows and three columns. The columns are:
- part_num: Unique ID for the part (string).
- name: Name of the part (string).
- part_cat_id: Part category unique ID (integer) (from `part_categories.csv`).

part_cat_id: the part category from `part_categories.csv.`

APPENDIX B.

Group Project Instructions

- For this assignment, use your Group Oracle Account. Each team member will have access to this account.
- There are seven deliverables for this project

Part 1 Group Contract (5 Points)

1. Complete and submit the Group Contract by 04/20 @ 11:55 PM via the course management system. An example Group Contract is attached. You may use the example or make one of your own.

Submit the group contract via the course management system. This is deliverable #1

Part 2 The Data Model (10 Points)

2. Go to https://www.kaggle.com/rtatman/LEGO-database and download the data.

3. There is much information about this data on this webpage. Please take the time to use the view information about each file in the dataset.

4. The Kaggle site has a rudimentary Entity Relationship Diagram ERD of the LEGO® dataset (https://www.kaggle.com/datasets/rtatman/LEGO-database?select=downloads_schema.png). Using any application you want, recreate the ERD using the crow's foot method we discussed in the first half

of the course. Include all attributes, primary keys, foreign keys, maximums, and optional/mandatory indicators.

Submit a PDF of your group's ERD. This is deliverable #2

Part 3 Creating and Loading the Kaggle Data (35 Points)

5. Drop all the tables mentioned in the above ERD (in case you already have tables with these names) EXCEPT inventory_sets. We will not be using inventory_sets for this project.
6. Use DDL to create the inventories, inventory_parts, parts, and sets tables
7. Load the data from Kaggle.com into the inventories, inventory_parts, parts, and sets tables
8. Write the DDL needed to create the colors, part_categories, and themes tables.
9. Load the data from Kaggle.com into the colors, part_categories, and themes tables.
10. Write the SQL needed to create the primary keys for each table.
11. Create all needed foreign keys. Please note that EVERY relationship in your ERD represents a foreign key.
12. SET ECHO ON and Run DESCRIBE on each table created for this project
13. Run SELECT * on each table created for this project. Show only the first ten rows from each table.

Submit a single file (plain text) showing the SQL and results for questions 5-13. This is deliverable #3

Part 4 Querying the Data (10 Points)

14. Query the proper systems/dictionary table to show all constraints on each table you created. Only show the tables you created for this assignment—order tables in ascending order by name. Use FORMAT and SET PAGESIZE as needed to improve the appearance of the results.
15. Create the query to answer: How many red parts are in the LEGO® data? Count all parts with red anywhere in the color name.
16. Create the query to answer: What are the Parts Categories with the highest percentage of spare parts compared to non-spare parts? Show the top 5 in descending order. Order by column number. Do not order by column name or alias.
17. Create the query to answer: What is the parent theme with the most "children" themes?
18. Find the average number of pieces in each LEGO® set (by year). Give the average number of pieces a meaningful alias. Order the results from highest to lowest using the alias for the average number of pieces. Show only the top 8 years.
19. Create the query needed to answer: Which set has the most unique spare parts?
20. Create the query to answer: Which theme has the most total parts across all sets? Show the name and the number of pieces. Show only the top theme (or themes if there is a tie) – not all.
21. What is/are the oldest sets in the LEGO® data WITH a Guardians of the Galaxy theme? Show only the oldest set (or sets if there is a tie) – not all. You must use a NATURAL JOIN for this question.

Submit a single file (plain text) showing both the SQL and results for questions 14-21. This is deliverable #4

Part 5 Entering New Data (30 Points)

22. Enter all data for LEGO® set 11966-1: (Pirate Ship) into your tables. More information about LEGO® set 11966-1: (Pirate Ship) can be found here: https://brickset.com/sets/11966-1
Enter all data via INSERT statements.
23. You must enter the set into the set table and populate all other tables as needed. There are 33 parts, but some are duplicates. Some may already exist in the data, but others must be added. This part may be the assignment's most time-consuming (and difficult) part. Please plan accordingly.
24. SET ECHO ON and Run the needed SELECT statements to show that you have loaded the data correctly. SHOW ALL DATA from #22/#23. Only show the data related to the Pirate Ship set. Use FORMAT and SET PAGESIZE if needed to improve the appearance of the results.

Submit a single file (plain text) showing the SQL to answer questions 22-24. This is deliverable #5

Part 5.a. – for graduate student teams only
You must also enter all data for an additional LEGO® set -- https://brickset.com/sets/11961-1/Helicopter

Repeat steps 22-24 for this LEGO® set -- include with deliverable #5

Part 6 Feedback (10 Points)
25. Write a short statement describing each member's contribution to the project. This is deliverable #6 (5 points)

26. Each group member must complete a survey on team member contributions. This is an individual assignment.

I will post the survey during Week 15. This is deliverable #7 (5 Points).