Summer 6-2021

Detecting Dirty Data Using SQL: Rigorous House Insurance Case

Daniel Street
das051@bucknell.edu

James G. Lawson
Bucknell University

Journal of Accounting Education 55 (2021) 100714

Contents lists available at ScienceDirect

Journal of Accounting Education

journal homepage: www.elsevier.com/locate/jaccedu

Detecting dirty data using SQL: Rigorous house insurance case

James G. Lawson, Daniel A. Street *

Bucknell University, Freeman College of Management, Lewisburg, PA 17837, United States

ARTICLE INFO

Article history:
Received 29 December 2018
Received in revised form 18 December 2020
Accepted 10 January 2021

Keywords:
Data analytics
Accounting education
Dirty data
Structured query language ("SQL")
Data integrity

ABSTRACT

Proficiency with data analytics is an increasingly important skill within in the accounting profession. However, successful data analysis requires clean source data (i.e., source data without errors) in order to draw reliable conclusions. Although users often assume clean source data, this assumption is frequently incorrect. Therefore, identifying and remediating "dirty data" is a prerequisite to effective data analysis. You, an accountant working at a firm that specializes in data analytics, have been hired by Rigorous House Insurance to analyze the company's claim insurance data. In addition to investigating specific issues mentioned by the company's controller, you are tasked with identifying any other data integrity issues that you encounter and providing preventative information system internal control suggestions to the client to mitigate these issues in the future.

1. The case

1.1. Project description

Rigorous House Insurance ("RHI") was founded by Dua Goodjob on January 1, 1950. It provides homeowners insurance for its customers' primary residences with the goal of providing reliable coverage and an excellent client experience. The company has been insuring homes in the southeastern United States since it began operations, but the current CEO of the company, Idida Goodjob, is looking to expand the company's market. Mr. Goodjob has recently hired a new Controller and has tasked the Controller with reviewing the company's operations to see if RHI is ready to expand its coverage outside of the southeast.

The Controller of RHI, Izzie Serious, has begun a thorough review of the company's financial standing, systems, and operations. Operational and financial reports that he has received recently have shown an upswing in the cost of house insurance claims relative to the total insured value. Upon looking into this issue, Mr. Serious became concerned about the possibility of errors in the corporate insurance claims data. Mr. Serious decided to hire your team of local accountants with expertise in data analytics to investigate the issue. Your local firm is known for its expertise in data analytics primarily due to an innovative eight-step process it has developed to detect dirty data (located in Section 1.4.). Mr. Serious has tasked you with analyzing recent insurance claim data to identify any data issues that appear to be present.

In the following discussion, table and field names will be indicated in parentheses, where relevant. As part of your initial investigation, you reached out to the Chief Operating Officer of RHI, Goahedin Paydem. Mr. Paydem provided you with an overview of the house insurance claims process, and responded to your questions. You asked whether the cost of an insurance claim (claim_data.requested_amount, claim_data.approved_amount, and claim_data.check_amount) was related to the

* Corresponding author.
E-mail addresses: james.lawson@bucknell.edu (J.G. Lawson), daniel.street@bucknell.edu (D.A. Street).

https://doi.org/10.1016/j.jaccedu.2021.100714
0748-5751/

value of the customer's home (customers.house_value). Mr. Paydem responded that the cost of an insurance claim is based on the value of the customer's house (customers.house_value) as well as the part of the house that is damaged (claim_data.part_of_house). He provided you with estimates of the minimum and maximum percentage of a home's appraised value (reasonable_ranges.bottom, reasonable_ranges.top, and customers.house_value, respectively) that could reasonably be claimed depending on the part of the house that is damaged (reasonable_ranges.part_of_house). You also asked Mr. Paydem about the quality of the corporate insurance claims data that RHI relies on. In response, Mr. Paydem acknowledged that while the corporate insurance claims data is very important to the company, he does not have a good understanding of how the database is managed. Thus, it is possible that there are errors in the data.

Mr. Serious made two specific requests of you: First, he has asked you to identify any claims for which the amount paid is outside of the reasonable range estimated by Mr. Paydem. Second, he has asked you to produce a list of customers whose total claim check payments (claim_data.check_amount) for the year (based on the claim_data.event_date) exceed the limits of their insurance policies (coverage_limits.policy_type and coverage_limits.coverage_limit). Both of these issues could be caused by poor internal controls. In addition to these two specific requests, Mr. Serious notes that RHI will not be able to expand profitably if the company cannot make reliable business decisions based on its corporate insurance claims data. Consequently, he wants you to identify any other issues that you encounter in the data. If there are any other issues, he would like you to provide the company with preventative information system internal control suggestions to stop these data integrity issues from occurring again in the future.

1.2. The insurance claims process

Mr. Paydem provided the following details to you regarding RHI's insurance claims process: "After customers experience an accident damaging a part of their home, they call us to initiate an insurance claim to compensate them for repairs. Customers provide RHI with the date of the accident (claim_data.event_date), information about what part of the house was damaged (claim_data.part_of_house), and the amount that they paid for repairs (claim_data.requested_amount). An RHI employee receives this information over the phone and enters it into the database. After entering this information, the employee assigns a claim ID (claim_data.claim_id) and enters the date the claim was received by RHI (claim_data.claim_submission_date) and the customer ID (claim_data.customer_id) into the database."

"Next, RHI inspectors evaluate the claim and compare the requested claim amount (claim_data.requested_amount) to the coverage limits of that customer's insurance policy type (coverage_limits.policy_type and coverage_limits.coverage_limit). The coverage limit for each insurance policy type is based on a customer's appraised house value (customers.house_value). Total payments (claim_data.check_amount) for accidents occurring within a year (based on the claim_data.event_date) may not exceed a given percentage of the value of the customer's house (claim_data.coverage_limit). The company offers three different insurance policy types (customers.policy_type): Basic, Complete, and Premium. For the 'Basic' policy type, total payments to a customer are limited to no more than 60% of the customer's appraised house value per year (based on the year of the accident event date). For the 'Complete' policy type, total payments to a customer are limited to no more than 100% of the customer's appraised house value per year. For the 'Premium' policy type, total payments to a customer are limited to no more than 120% of the customer's appraised house value per year. After the claim is evaluated, RHI employees record the amount of the claim that should be paid (claim_data.approved_amount) as well as the date that decision was made (claim_data.claim_decision_date). Finally, after a check is mailed to the customer for the claim, RHI employees enter the amount (claim_data.check_amount) and date of the check (claim_data.check_date) into the database."

1.3. The data

The 2017 insurance claim data for RHI has been provided to you. The data dictionary in Table 1 describes each table in the database. The fields in each table of the database are described in separate rows. The IT personnel of RHI did not state whether any primary key, foreign key, or field type constraints are present in the database.

1.4. Firm resources for data analysis

1.4.1. Effective database design and control

Effective database design and control ensures data integrity as data proceeds through three phases: input, processing, and output (Chapter 10, "Processing Integrity", Romney and Steinbart, 2018). Data input is the process of creating or entering data into an information system. Internal controls such as proper input form design, edit checks, check digits, and segregation of duties should be in place to ensure that input data is valid, complete, and accurate. In a relational database, the database management system should include the following list of input controls: primary keys should be unique, foreign keys should match to the primary key of another table, fields should only accept data of the appropriate type (numeric, character, date, etc.), records should not be committed to the database while any required data is missing, limit and reasonableness checks should prevent obviously incorrect numeric data from entering the database. Data processing consists of reading, updating, or deleting data in an information system. Internal controls over data processing such as batch or hash totals, data validation, and write-protection mechanisms should be in place to ensure that data is updated accurately and that data processing does not cause any errors in stored data. Finally, output is the process of extracting data from the information system

Table 1
Data dictionary.

Panel A: Claim_Data (11 Fields, 1005 Records)

Field	Definition
claim_id	A number that identifies each claim
customer_id	A number that identifies each customer
claim_status	The current status of the claim in the review and approval process; one of four values: "1.Submitted", "2.Under Review", "3. Approved", and "4.Paid"
part_of_house	The part of the house that was impacted by the accident; one of six values: "Deck", "Garage", "Plumbing", "Roof", "Siding", or "Window".
event_date	The date of the accident
requested_amount	The amount requested by the customer for the claim
claim_submission_date	The date that the claim was submitted to RHI for payment
claim_decision_date	The date that RHI approved or denied the claim
approved_amount	The amount that RHI agreed to pay for the claim (provided that the claim was approved)
check_date	The date that a check was sent to the customer to pay for the claim
check_amount	The amount of the check sent to the customer to pay for the claim

Panel B: Customers (7 Fields, 178 Records)

Field	Definition
id	A number that identifies each customer
name	The customer's name
policy_start_date	The start date of the customer's policy
policy_end_date	The end date of the customer's policy
policy_type	The insurance policy type purchased by the customer; one of three values: "Basic", "Complete", or "Premium"
birthdate	The customer's birthdate
house_value	The most recent appraised value of the customer's home

Panel C: Coverage_Limits (2 Fields, 3 Records)

Field	Definition
policy_type	The name of each insurance policy type offered; one of three values: "Basic", "Complete", or "Premium"
coverage	The maximum claim payout permitted by the insurance policy type in a given year, expressed as a percentage of the customer's house value

Panel D: Reasonable_Ranges (3 Fields, 6 Records)

Field	Definition
part_of_house	Text which uniquely identifies each part of the house covered by the insurance policy; one of six values: "Deck", "Garage", "Plumbing", "Roof", "Siding", or "Window".
bottom	An estimate of the minimum percentage of the customer's house's appraised value that could reasonably be claimed for an accident
top	An estimate of the maximum percentage of the customer's house's appraised value that could reasonably be claimed for an accident

to prepare reports, documents, or other information displays. Internal controls over output such as reconciliations, user access control, and secure document disposal should be in place to ensure that the data is accessible only to users authorized to receive the output.

Although these are the principles for effective database design and control ensure data integrity, in practice firms do not always adhere to these principles. A lack of effective database design and control can lead to inaccurate, incomplete, or invalid data ("dirty data"). The firm also has provided you with an eight-step process to detecting dirty data.

1.4.2. Firm guide to detecting dirty data

Step 1: Obtain an understanding of the business process represented by the data. This can be accomplished by asking involved personnel or reading industry standards. This first step is critical, because understanding the business process will guide you towards learning what elements the data should ultimately capture.

Step 2: Obtain an understanding of the source of the data. How is the data entered into the database? Is it electronically transmitted? Scanned in via a barcode? Manually keyed in? When evaluating how the data is entered into the database it is important to consider Murphy's Law: anything that *can* go wrong *will* go wrong. You should ask questions like: What could go wrong in the data entry process? What edit checks are applied to the data entry? Are there any known periodic reviews of the data? Although taking the view that anything that can go wrong will go wrong may seem pessimistic, this critical thinking approach will allow you to begin to identify how dirty data could enter the system. Cleansing the data set begins with an understanding of its potential faults.

Step 3: Obtain any available data dictionaries, database diagrams (schemas), or system flowcharts to enable a more detailed understanding of the fields and relationships in the data. Internal company analysts, IT professionals, or internal auditors may be helpful sources of this information.[1] These resources help provide you with a sense of the 'lay of the land' of the database and provide them with valuable information regarding the structure of the data. If there are *not* data dictionar-

[1] Even though these resources are incredibly valuable in understanding the underlying data set, many companies will not have a data dictionary or database diagram for the data set being analyzed. Regardless of whether these resources are available, it is a best practice to request them.

ies, database diagrams, or systems flowcharts available, then you should develop your own expectation of the elements that the data should contain. For example: What tables should be present? What might be the primary and foreign keys? What variables and variable types should be located in the data?

Step 4: Review the most recent 100–250 rows of data to gain additional familiarity with the data and the business process. This review may immediately identify some issues in the recorded data. Even if there are no immediately obvious issues, reviewing the data should enhance your understanding of the data and business process at hand.

Step 5: Generate and review summary reports for the data. These summary reports can identify several dirty data issues. Across all field types, summary reports may identify the extent of missing data for each field. Summary reports of numeric and date variables may provide minimum, average, and maximum amounts and dates. Summary reports of non-numeric fields may be used to determine the most and least common entries in a categorical field.

Step 6: Develop and note specific expectations about the data *within* each field. Expectations should be generated for all types of data: primary and foreign keys, amounts, dates, statuses, categories, etc. Here are three specific expectations that are likely to apply in most datasets:

a. Primary keys for a table should not be missing and not be repeated
b. Foreign keys in a table should not be missing.
c. Statuses and other categorical fields should contain only valid entries [2]

Step 7: Develop and note specific expectations about the relationships that should hold *between* fields. Expectations should be generated for all types of data: primary and foreign keys, amounts, dates, statuses, categories, etc. Here is one specific expectation that is likely to apply in most datasets:

a. Foreign keys in a table should match the primary key of another table[3]

Step 8: Write exception reports to test each of the specific expectations identified in Steps 6 and 7. Any exceptions noted by these reports reveal cases in which your expectations have not been met. Some exceptions may simply indicate that you need to improve your understanding of the business and data entry processes that create the data. Other exceptions will identify dirty data and will show you what elements of the data need to be addressed by remediation efforts.

In addition to providing you with a more robust understanding of the underlying data and the business processes that created the data, this eight-step process will help to ensure that subsequent data analytic procedures are based upon reliable source data.

1.4.3. The data analysis process

Data analysis may begin once data is stored and processed in an information system. Firms often utilize Extract, Transform, and Load (“ETL”) software to retrieve, process, and load data into information systems. The first phase of a data analysis project is to form an objective (e.g., answer a question, test a hypothesis, optimize an outcome). The second phase is to identify and collect the data that will be necessary to achieve the objective. The third phase is to assess the quality of the data collected (the data’s validity, completeness, and accuracy) and to remediate any problems identified within the data (sometimes referred to as “cleaning” the data). The fourth phase is to use analytical methods in support of the objective (e.g., summing, averaging, identifying, classifying, predicting). Finally, the fifth phase is to interpret the results of the analysis and to communicate those results as appropriate. If a data analysis project is conducted based on *invalid, incomplete, or inaccurate* data (i.e., if the data cleaning process is ineffective), then the calculation of analytical results in the fourth phase will be incorrect. If incorrect analytical results are communicated to stakeholders, then those stakeholders arrive at the wrong conclusion and make poor decisions, harming themselves and others.

1.5. Project requirements

As you review the RHI claim data, you discover that Mr. Serious’s concerns are well founded. Guided by your accounting firm’s eight-step process for detecting dirty data, you discover a large number of issues with the data. There are at least twenty-four distinct problems in the claim data you have received. Based on the 2017 insurance claim data provided, please work through your accounting firm’s eight-step process for detecting dirty data (located in Section 1.4.2.), complete the following tasks, and provide the following two deliverables.

1) Prepare a memo responding to Mr. Serious containing the following information:

a) A well-formatted entity relationship diagram.

[2] Other expectations regarding the data within each field may include the following:X amount should be no more than NX date should be no earlier/later than NX field should have only numeric (character) dataThere should be no null values in X fieldAll values in X field should be unique or sequential

[3] Other expectations about the relationships that should hold between fields may include the following:Records with X status should have data present in Y amount and Z date fieldsX date should be before the dates in Y and Z fieldsX amount should be no more than the amount in Y fieldField X in the transactional data table should match Field Y in the master data table

b) A list of expectations to be verified within single fields (e.g., no duplicate Customer IDs). Refer to Step 6 of Section 1.4.2.
c) A list of expectations about logical relationships to be verified between fields (e.g., total claims paid for the year is not greater than permitted under the customer's policy). Refer to Step 7 of Section 1.4.2.
d) Based on the expectations you developed in Requirements 1b and 1c, identify at least five distinct problems present in the claim data in addition to the two specific requests made of you by Mr. Serious. Refer to Step 8 of Section 1.4.2.
e) Describe each of the five problems you identified in Requirement 1d in a paragraph. In the paragraph describing each problem, propose a preventative information system internal control to mitigate the extent of the problem in the future. How would you advise Mr. Serious to proceed to resolve these problems?
f) How would you advise Mr. Serious to proceed to resolve the two concerns he noted at the end of Section 1.1.?

2) Using the provided database, write the following queries. Provide the text of your queries and a screenshot of the output in a separate document:

a) A query identifying the smallest claim payment (claim_data.check_amount) in 2017.
b) A query identifying the birthdate (customers.birthdate) of the most recently born customer.
c) A query identifying each different policy type (customers.policy_type) in the customer data.
d) A query responding to Mr. Serious' first request: Identify any claims for which the amount paid (claim_data.check_amount) is outside of the reasonable range (reasonable_ranges.bottom and reasonable_ranges.top) estimated by Mr. Paydem. As discussed in Section 1.1., the reasonable range estimated by Mr. Paydem (reasonable_ranges.bottom and reasonable_ranges.top) is based on the customer's house value (customers.house_value) and the part of the customer's house that is damaged (claim_data.part_of_house and reasonable_ranges.part_of_house).
e) A query responding to Mr. Serious' second request: Identify customers by name (customers.name) whose total claim check payments (claim_data.check_amount) for the year (based on claim_data.event_date) exceed the limits of their insurance policies (coverage_limits.policy_type and coverage_limits.coverage_limit). As discussed in Section 1.2., a customer's insurance policy coverage limit is based on the customer's appraised house value (customers.house_value) and their insurance policy type (customers.policy_type).
f) Queries and listings of the records affected by each of the five additional problems that you documented in Requirement 1e.

2. Teaching notes

The remaining contents of the paper are as follows: In Section 2.1, we provide the motivation for our case. In Section 2.2, we provide an overview of related teaching cases and discuss how our case differs from prior cases. In Section 2.3, we discuss the learning objectives for the case, and we provide evidence of the efficacy of the case in achieving the learning objectives in Section 2.4. In Section 2.5, we provide implementation guidance. We conclude in Section 2.6 by discussing where students succeeded and where students struggled while completing the case. In Appendix 1, we provide a full list of the case files and instructor resources that are available for this project upon request from the authors. We also provide a short description of each instructor resource in Appendix 1.

2.1. Motivation: Accountants in a data-driven world

The ability to work with and analyze large data sets is growing from a niche specialty to a core skill required by almost all accounting professionals. Data analytics is now used by auditors as they continuously audit their clients (Zhang, Yang, & Appelbaum, 2015), tax accountants as they assist their clients with tax planning (DaBruzzo, Dannenfelser, & DeRocco, 2013), and financial and managerial accountants as they prepare financial reports (Schneider, Dai, Janvrin, & Raschke, 2015). Large accounting firms are eager for new accountants to be well versed in data analysis (PWC, 2015), and KPMG has gone so far as to partner with certain universities to offer a Master of Accounting with a Data and Analytics focus (McCabe, 2018). The demand—and need—for accountants who understand the principles of successful data analysis has never been higher.

Because proficiency with data analytics is so important for accountant graduates, accounting educators have a responsibility to equip their students with these skills. While many accounting programs are attempting to integrate data analytics into their accounting curriculum, challenges still exist. For example, Igou and Coe (2016) reviewed eight Accounting Information System textbooks and found that the textbooks normally did not include any data analytics projects. While there have been a variety of data analytics cases published in accounting education journals (see the following section for a full review), there is still no case that teaches students one of the most fundamental skills necessary for successful data analytics: how to clean the data. Of the five phases of a data analysis project (1 - form an objective, 2 - identify and collect data to achieve the objective, 3 - assess data quality and clean the data, 4 - use analytical methods to achieve the objective, and 5 - interpret and communicate the results of the analysis), cleaning the data is one of the most time-consuming steps. A recent survey of data scientists revealed that approximately 60% of the time spent working with data was spent cleaning

and organizing it (Press, 2016). [4]Another estimate places the time spent preparing the data as 60–75% of the total time on the project, and cautions that ignoring the quality issue at the beginning can result in a significant amount of rework later on (Sherman, 2015). Identifying "dirty data" is a foundational aspect of data analysis because valid, complete, and accurate source data is a prerequisite for successful data analysis (AICPA 2020; ISACA 2018).

We define "Dirty data" as a data set that is invalid, incomplete, or inaccurate (AICPA 2020; ISACA 2018).[5, 6]Most data analytics cases—and most claims about how data analytics can be used in the accounting profession—are built on the assumption that the underlying data set is accurate and complete. However, this assumption frequently does not reflect reality. A 2015 survey by Experian revealed that corporations estimate that 32% of their data is affected by inaccuracies (Haselkorn, 2015). Any conclusions drawn from analysis of dirty data sets will be biased and the severity of the data integrity issues will determine the inaccuracy of the conclusions drawn. One cause of dirty data is customers or employees entering incorrect information into a database. However, human transcription errors are not the sole source of dirty data—system design flaws can also result in incomplete or inaccurate data. Accounting professionals need to be knowledgeable regarding how to identify and remediate dirty data. Any analysis performed on a dirty data set is not fully reliable, be it continuous auditing or revenue forecasting. Consequently, ensuring that underlying data sets are "clean" should always be the first step in any data analytics project. This case is designed to prepare accounting students to detect dirty data problems as they prepare to enter a Big Data world.

2.2. Related teaching cases

The accounting education literature contains a number of cases that deal with big data and data analytics. The majority of these cases deal with data visualization and subsequent analysis (Janvrin, Raschke, & Dilla, 2014; Igou & Coe, 2016; Kokina, Panamanova, & Corbett, 2017; Cunningham & Stein 2018).[7] These cases require students to use a particular software (normally Excel or Tableau) to visualize data and draw inferences to apply to the set of challenges or opportunities faced by the business organization in the case. Another case (Angelo, Ayres, & Stanfield, 2018) focuses on the procedures that can be used to analyze large data sets. Although these cases equip students with the ability to visualize and analyze large data sets, our case contributes to the literature by focusing solely on the first step of successful data analysis—cleaning the data.

Notably, there are two cases dealing with data analytics that focus to some extent on the quality of the underlying data. Enget, Saucedo, and Wright (2017) introduce a case entitled "Mystery, Inc." that requires students to use big data to identify anomalies in the context of journal entry testing. "Mystery, Inc." is exclusively focused on journal entry testing, and the data anomalies present in that case (e.g., unusual user activity, journal entries made during non-business hours, journal entries with missing descriptions, and notable revenue/expense trending) are more indicative of potential fraud than anomalies caused by dirty data. While "Mystery, Inc." focuses on data anomalies which suggest fraud, we focus on data anomalies outside of the fraud context which may still hinder effective data analysis.

"Manual Journal Entry Testing: Data Analytics and the Risk of Fraud" (Fay & Negangard, 2017) is another case focused on data analytics and journal entry testing with an emphasis on analytic procedures that reveal the possibility of fraud. Notably, Fay and Negangard acknowledge that one of the primary challenges of working with big data is that the data must be validated prior to performing any analysis (p. 39). As a result, the first phase of their case requires students to validate the data set by using IDEA to ensure that the data file received from a client is complete. The notion that the underlying data must be validated ("cleaned") before any successful analysis can be completed is the cornerstone of our case. Our case is similar to the case introduced by Fay and Negangard (2017) in that students have to make sure the data is clean before conducting any analyses. However, the only criteria for clean data in "Manual Journal Entry Testing: Data Analytics and the Risk of Fraud" is that the data be complete, whereas our case introduces a more comprehensive understanding of clean data. For example, our case teaches students to develop expectations regarding logical relationships within fields (e.g., no duplicate primary keys) and between fields (e.g., the date a claim is submitted by the customer should precede the date that a check is issued to pay for the claim). We also teach students to detect dirty data issues by writing SQL queries and to suggest preventative information system internal controls that would stop dirty data issues from recurring.

[4] Firms will often utilize extract, transform, and load ("ETL") software to retrieve and consolidate data (Vassiliadis 2009). This process is normally not conducted by accountants, but understanding how underlying datasets are created is a necessary part of the data analysis process. Although ETL processes are generally automated, the eight-step process to detect dirty data introduced in this paper could be integrated into an ETL process. Specifically, the eight-step process could be used to design exception reports which are then automatically generated as part of the ETL process. Although producing these exception reports will not remediate dirty data issues on its own, these reports could be provided to individuals with the necessary expertise to address the issue. We note that COBIT 5 DSS06.02 Activity 3 recommends that data needing correction be sent back "as close to the point of origination as possible", so exception reports that identify problems with input data should be sent back to the individuals or systems which introduced that dirty data to the database.

[5] Common examples of dirty data include missing data, duplicate data, data not in the expected format, unreasonable amounts, dates, and category values, data that fails to follow expected business rules, and transaction data that is inconsistent with master data.

[6] The objective of processing integrity in the Trust Services Framework and COBIT 5 DSS06.02 also include the data characteristics "timely" and "secure" (or "authorized"). Although these data characteristics are undoubtedly important, this case focuses on identifying data which is invalid, incomplete, or inaccurate, regardless of cause. If input data are not recorded and updated *timely*, the resulting data will be invalid, incomplete, or inaccurate. Similarly, if data, at any point in the data processing cycle, are *insecure* or open to *unauthorized* changes, then the data may be subject to insertion, alteration, or removal. This could result in the data becoming invalid, incomplete, or inaccurate.

[7] One case that does not neatly fall into the visualization or analysis category is "Data Governance Case at KrauseMcMahon LLP in an Era of Self-Service BI and Big Data" (Riggins and Klamm 2017), which is focused on corporate governance issues related to data policies.

While using SQL queries to identify dirty data issues is a unique contribution to the accounting education literature, several prior accounting cases do deal with queries and relational databases. These cases require students to use queries to conduct analyses (Zanzig & Tsay, 2004; Loraas & Searcy, 2010) and emphasize the value of relational databases over traditional spreadsheets (Borthick, Schneider, & Viscelli, 2017). We expand this literature by teaching students to create queries that can be used as detective internal controls. Additionally, prior cases assume that the data stored in relational databases is clean and therefore provide students with clean databases to work with. We provide students with a dirty data set to emphasize that, in practice, accountants will frequently encounter data that is less than perfectly clean. Our case equips students with the ability to detect dirty data so that appropriate steps can be taken to remediate these issues, ultimately leading to more accurate and reliable analyses.

2.3. Learning objectives

The learning objectives for our case, which we have mapped onto Bloom's Revised Taxonomy (Anderson, Krathwohl, & Bloom 2001), are as follows:

1. Learn to create and understand exception reports using SQL queries to identify dirty data in the provided dataset.
2. Learn, remember, and understand five examples of data integrity errors, and apply SQL queries to identify them in the provided dataset.
3. Evaluate the current process of data collection and management in order to identify data control issues, then create preventative information system internal controls to ensure data integrity in the future.

In 2018, AACSB International updated its standards for separate accreditation of accounting programs. Dzuranin, Jones, and Olvera (2018) argue that many business schools face a sense of urgency in attempting to comply with AACSB accreditation standards, and we specifically designed our case to address the updated AACSB curriculum standards. AACSB Standard A5 details the curriculum requirements related to information technology in accounting and business. The standard lists three primary curriculum components to be covered by accredited institutions. The first component deals with data creation, manipulation, security, and storage; the second component deals with data analytics; the third component deals with "information technology agility" and the need for continual learning of relevant skills.

This case integrates all three components of the 2018 AACSB's information technology standards. Learning Objective 3 satisfies component one as it requires students to suggest improvements for a database. Learning Objectives 1 and 2 satisfy component two as they deal with cleansing dirty data, which is a prerequisite to successful data analytics. Finally, Learning Objectives 1–3 all satisfy the third component of Standard A5 as they require students to develop relevant skills that will help them succeed in their careers.

2.4. Case efficacy

The Rigorous House Insurance Case has been taught in five upper-level accounting classes at four different colleges—Accounting Information Systems classes, Data Analytics classes, and an accounting elective that had a focus on data visualization and analytics. To demonstrate the teaching efficacy of our case, we administered both a knowledge test and a survey assessing student proficiency to students in two of those five classes (17 students in an undergraduate accounting information systems course at a large western comprehensive university in spring 2019 and 76 students in a master's level accounting elective with a focus on data analytics and visualization at a large southeastern research university in spring 2020). All student responses were collected anonymously.[8] Neither of the authors were instructors of the classes where efficacy data was collected. Efficacy data collected from these two schools is provided in Table 2.

This case has also been taught in undergraduate accounting information systems courses at a large southeastern research university in spring 2018 and a large midwestern research university in fall 2018. An anonymous reviewer administered this case in an accounting data analytics class for undergraduate and graduate students at an anonymous university during spring 2019 and a guest editor administered this case in an undergraduate accounting class in fall 2020[9]. Although formal efficacy data was not collected from the other times this case was taught, the instructors indicated positive experiences with the case.

We report the results from the pre- and post-case knowledge test in Table 2, Panel A. Students completed this knowledge test before receiving any case materials and completed the knowledge test again after finishing the case study. The four knowledge test questions are directly related to the learning objectives of this case. Specifically, questions 1 and 2 are related to Learning Objective 1, question 3 is related to Learning Objective 3, and question 4 is related to Learning Objective 2. In Table 2, Panel A, we display t-tests comparing student responses on the pre- versus post-tests (p-values are presented and asterisks denote statistically significant changes). There is no significant difference in student responses for question 1. However, questions 2–4 all show a significant improvement in student performance when comparing the pre and post

[8] The efficacy data was collected anonymously via survey. Of the 93 total, eight students completed the pre-case survey but not the post-case survey.

[9] We are grateful to the anonymous reviewer and the guest editor for providing us with valuable feedback about their experiences with this case.

Table 2
Efficacy.

Panel A: Pre-case versus post-case knowledge test **Question, Correct Answer, *Incorrect Answer Choices***	**Pre n**	**Post n**	**Pre rate correct**	**Post rate correct**	**Significance of difference**
1. Which of the following is used to retrieve data from a database? Query, *Schema, Table, Record*	93	85	0.978	0.988	p = 0.609
2. Which of the following terms is not a SQL statement? Retrieve, *Select, From, Where*	93	85	0.957	1.000	p = 0.041**
3. Which of the following controls prevents an input error from being recorded within the database? Edit check, *Exception report, User manual, Batch processing*	93	85	0.785	0.941	p = 0.002***
4. Which of the following is most likely to represent dirty data? Two transactions reference the same customer ID, *Two customer records reference the same customer ID, Two transactions reference the same product ID, One transaction references two product IDs*	80	82	0.860	0.965	p = 0.011**

Panel B: Pre-case student proficiency **Statement**	**Mean**	**SD** (n = 93)	**Significance** (Difference from 3 – Average)
1. My working knowledge of internal controls is:	3.129	0.783	p = 0.116
2. My working knowledge of database structures is:	2.753	0.747	p = 0.002***
3. My working knowledge of creating SQL queries is:	2.785	0.976	p = 0.036**
4. My working knowledge of how to detect data integrity problems is:	2.624	0.765	p < 0.001***

Panel C: Post-case student improvement in proficiency **Statement**	**Mean**	**SD** (n = 85)	**Significance** (Difference from 3 – Neither Agree nor Disagree)
1. This case increased my working knowledge of internal controls:	3.682	0.941	p < 0.001***
2. This case increased my working knowledge of database structures:	3.953	0.739	p < 0.001***
3. This case increased my working knowledge of creating SQL queries:	4.306	0.831	p < 0.001***
4. This case increased my working knowledge of how to detect data integrity problems:	4.035	0.837	p < 0.001***
5. This case enhanced my ability to identify deficiencies in internal control:	3.882	0.762	p < 0.001***
6. This case enhanced my ability to propose internal controls to mitigate risks:	3.871	0.737	p < 0.001***
7. This case helps me think more critically about limitations to big data analysis:	3.894	0.831	p < 0.001***
8. I think that the time devoted to the case was worthwhile:	3.529	1.019	p < 0.001***
9. I think analyzing data will be important to my future career:	4.318	0.790	p < 0.001***
10. I found this case interesting:	3.659	1.053	p < 0.001***

Scale: 1 = Very Poor, 2 = Below Average, 3 = Average, 4 = Above Average, 5 = Excellent.
Scale: 1 = Strongly Disagree, 2 = Somewhat Disagree, 3 = Neither Agree Nor Disagree, 4 = Somewhat Agree, 5 = Strongly Agree.
* Significant at the 10% significance level.
** Significant at the 5% significance level.
*** Significant at the 1% significance level.

responses (all p-values < 0.05). Our knowledge test provides evidence that the learnings objectives of this case were achieved.

We also conducted surveys of student proficiency related to the case study topics. We present the results from these surveys in Panels B and C of Table 2. Students took the survey in Panel B before receiving any case materials and took the survey in Panel C after finishing the case study. In Panel B, we report the results of students rating their proficiency in four areas related to the case topics. The survey responses range from "Very Poor" to "Excellent" on a 1 to 5 scale. We test for a difference between the student responses and an "Average" proficiency rating. Students reported an average proficiency in regards to internal controls. Notably, though, student assessment of proficiency related to database structures, SQL queries, and identification of data integrity problems was significantly lower than an average proficiency (means of 2.753, 2.785, and 2.624, respectively). The results in Panel B suggest that students were aware of a lack of working knowledge related to skills used to identify and prevent dirty data.

Students also responded to a survey regarding whether the Rigorous House Insurance Case improved their proficiency in several key areas. Students took this survey after completion of the case, and we report the results in Panel C. The survey responses range from "Strongly Disagree" to "Strongly Agree" on a 1 to 5 scale. We test for a difference between the student responses and a "Neither Agree Nor Disagree" rating. Students agreed that the case increased their working knowledge of internal controls, database structures, creating SQL queries, and detecting data integrity problems (means of 3.682, 3.953, 4.306, and 4.035, respectively). Students also agreed that the case was interesting and that time devoted to the case was worthwhile (means of 3.659 and 3.529, respectively).

Taking the results of all the efficacy tests together, the following picture emerges. Prior to completing the case study, students reported below average proficiencies related to dirty data issues (Table 2, Panel B). Following completion of the case, students indicated that the case study was a valuable learning activity that improved their working knowledge of dirty data issues (Table 2, Panel C). The changes in student performance on the knowledge test taken before and after the case study

(Table 2, Panel A) further indicate that student performance improved following the case. The results in Table 2 lead us to conclude that the case is effective at accomplishing the learning objectives.

2.5. Implementation guidance

While this case has been used in both accounting information systems and accounting data analytics classes, we strongly recommend covering query development via SQL in Microsoft Access or another comparable database management software before administering the case. Students should be taught about preventative information system internal controls and general database concepts prior to beginning this case. We also recommend spending one class session covering the eight-step process for identifying dirty data in detail. This eight-step process, provided in Section 1.4.2., is the foundation for the case and provides a roadmap for the students as they complete the assignment.

After teaching through the eight-step process, instructors should give students the case overview and the database file.[10] We also recommend that instructors provide students with a list of preventative information system internal controls or refer students to the relevant portion of their textbook.[11] The first three steps of the eight-step process correspond to certain portions of the student case material. Students obtain an understanding of the business process represented by the data (Step 1) via Section 1.1. Students obtain an understanding of the source of the data (Step 2) via Section 1.2. Finally, students obtain a data dictionary (Step 3) in Section 1.3. Students then complete Steps 4 to 7 based on the data provided. Students' work on these steps enables them to complete case requirements 1a to 1c and 2a to 2c.

To provoke students' curiosity and enthusiasm for the case as they develop expectations of the data in Steps 6 and 7, we recommend that instructors show students a few examples of the "dirty data" within the case data. In one case, the instructor showed examples of duplicate customer IDs (customers.id) and customers with policies starting (customers.policy_start_date) before their birthdates (customers.birthdate). After seeing these examples of unreasonable data, students became very interested in identifying similar issues. Additionally, this has the benefit of showing students an example of the type of distinct problems that they are required to identify.[12]

Once students have the case overview, database file, and have completed Steps 1 to 7, students can begin Step 8 – writing exception reports via SQL queries to identify instances of dirty data. Although there are at least 24 known dirty data problems in the dataset, we require students to identify only five distinct problems in case requirement 1e for efficiency. [13]This quantity can be increased or decreased according to the instructor's discretion. Students' work on Step 8 enables them to complete case requirements 1d, 1e, and 2d to 2f.

We require individual students[14] to write SQL queries as part of the case deliverable for a variety of reasons: First, SQL provides a variety of commands such as INNER JOIN or LEFT JOIN that easily relates different data elements. Second, SQL is standard across a variety of software, such as Tableau, SAS, and R.[15] Third, SQL can be used with data sets of virtually unlimited size whereas traditional spreadsheets will often crash if a worksheet has too many records and formulas.[16] We recommend giving students two weeks to complete the assignment, and we estimate that it took the students 15–20 hours to complete the assignment.

[10] Case data is available in Microsoft Access, Microsoft Excel, or SQLite form. Instructors may wish to provide students with the data in Microsoft Excel to teach students to import data and create a database in Microsoft Access. One limitation of using Microsoft Access is that students with Macs will not have the program on their laptop. An alternative is to use the program TablePlus, available at https://tableplus.io/. TablePlus is available for Windows, Mac, and iOS. TablePlus allows students to execute SQL queries and manage a wide variety of database structures including MySQL, SQLite, PostgreSQL, Microsoft SQL Server, and Amazon Redshift. Students who cannot or do not wish to install either Microsoft Access or TablePlus can use web-based SQL tools including SQL OnLine IDE (www.sqliteonline.com) or DB Fiddle. Our case data has been preloaded and is available on the latter tool at https://www.db-fiddle.com/f/6HxGYiyR5Ejyisr1vn5ujb/0. We thank an anonymous reviewer for suggesting the use of web-based SQL tools. For a full list of the resources available to instructors interested in administering this case, see Appendix 1.

[11] See, for example, the preventative information system internal controls listed and discussed in Chapter 10: Processing integrity and availability controls (Romney and Steinbart 2018).

[12] To assist instructors administering the case, we provide several other examples of the expectations that students should develop as part of Steps 6 and 7: 1) the approved amount (claim_data.approved_amount) should fall within the reasonable estimated range, 2) the approved amount should neither be negative nor greater than the requested amount (claim_data.requested_amount) 3) the approved amount should not exceed the amount covered by the customer's plan (claim_data.coverage_limit), 4) fields such as (claim_data.claim_submission_date, claim_data.event_date, claim_data.part_of_house, etc) should not have null values and 5) claim submission dates should not occur earlier than the date the company was founded; 6) claim submission dates should not occur before the event date.

[13] An instructor database key containing SQL queries satisfying requirements 2a through 2e as well as queries identifying each of the 24 known dirty data problems in the dataset can be obtained from the authors. For a full list of the resources available to instructors interested in administering this case, see Appendix 1.

[14] Alternatively, one instructor who administered the case had success by assigning the case as a group project with 2–3 students per group.

[15] An alternative to having students write their own SQL queries is to have students use Access's Query By Example ("QBE") tool. While this could be a beneficial starting place for students who are not proficient with creating queries, the QBE tool is not implemented identically in other software, so a student who only uses QBE to create queries may initially be at a loss when working in software such as Tableau, SAS, or R.

[16] Using a traditional spreadsheet to emulate the function of a relational database can be cumbersome, challenging, and resource-intensive. If one is working on a transaction dataset, for instance, and a foreign key is provided to link to the master data (e.g., customer number), one must use lookup functions to return every desired master data element. As worksheets grow in length and complexity, these lookup functions burden the RAM and processors of personal computers. Traditional spreadsheets are designed to organize data in just two dimensions: columns and rows. Relational databases, on the other hand, enable many different relationships between tables. Relational databases are optimized to enable the retrieval and aggregation of relational data (e.g., transaction and master data) and do so very efficiently. Borthick, Schneider, & Viscelli, 2017 provide a deeper comparison of traditional spreadsheets and relational databases.

After students turn in their deliverables, instructors can dedicate a portion of class time to conducting a debriefing session to emphasize important takeaways from the case.[17] Instructors can begin by discussing why cleaning the data is a prerequisite for successful data analysis. Instructors should also emphasize that the SQL reports developed during the case are only detective internal controls, not preventative or corrective, and that the process of correcting the issues identified will depend on the particular issue at hand (for example, a data issue caused by an error with an employee's data entry will have a different solution than an automated data processing error that creates duplicate observations). Instructors can also emphasize that developing specific expectations about the data within each field and the relationships between fields will form the foundation for creating exception reports that identify dirty data. Finally, instructors can ask students to describe which data issues they were most of proud of finding and why they looked for those particular issues. In our experience, students enjoyed telling the instructor how they were able to "play detective" and find unique errors.

Although the evidence suggests that students completing the entire case gain a rich and understanding of dirty data problems, this case may be subset, enabling students to learn some of the core concepts of the case without requiring them to complete the entire case. Here are two case subset options to consider:

1) **Develop expectations and create SQL queries:** Students are asked to develop and test their expectations of the data using Requirements 1a – 1d and 2f, but are not required to describe the dirty data problems or propose internal controls to address them in a memo (Requirements 1e-1f). The instructor may choose to make Requirements 2a-2e optional, although students often find that developing the queries in Requirements 2a-2e reveals several dirty data issues.
2) **Evaluate and resolve the client's concerns:** Rather than develop and test their own expectations of the case data set, students are asked to focus on evaluating and resolving their client's two concerns (provided at the end of Section 1.1.). Students are asked to complete Requirements 1a, 2d, 2e, and 1f.

2.6. Student performance

Overall, our efficacy results indicate that the students' knowledge improved after completion of the case. When providing feedback on the case, students generally reported that they felt as though they learned skills from the case that would be useful as they entered the accounting profession. Specifically, students felt they had gained an awareness of the presence of dirty data and the necessity of clean data for successful data analysis. They also agreed that this case increased their working knowledge of detecting data integrity problems. As described in Appendix 1, we offer several examples of student submissions as a resource to instructors.

While the students generally provided feedback that the case was a valuable learning activity, they also provided consistent feedback that the case was challenging. Very few of the students had experience writing SQL queries. To this end, and following feedback we received from other instructors who used the case in their classes, we recommend that instructors devote at least one class period to introducing SQL queries prior to beginning the case. Instructors can use this case data to work through sample SQL syntax and queries. These sample SQL queries can serve as tools that students use to complete the case assignment, and using this case data in class will allow students to become comfortable with the Rigorous House data. We have created a Powerpoint presentation that introduces basis SQL syntax and queries using the Rigorous House data as a resource for instructors.

Additionally, there are several helpful external resources instructors can use to introduce SQL queries to students. Sections 2-8 of the modules found at http://www.sqltutorial.org/ provide guidance on filtering data, joining tables, and grouping data, all of which can be used to identify dirty data. The first twelve lessons found at https://www.quackit.com/sql/tutorial/ provide similar instruction. Another excellent resource is "SQL: An Introduction to Writing Database Queries" by Dr. R. Drew Sellers (Sellers, 2017).[18] GalaXQL is a gamified, interactive SQL tutorial available at http://sol.gfxile.net/galaxql.html. Once students have experience with SQL query functionality — mastery is not required — they will be equipped to create queries that identify instances of dirty data and will be able to complete the assignment.

Students identified a wide variety of the dirty data issues seeded into the case. Students were able to easily identify problematic null values (customers.house_value, customers.id, claim_data.check_date, and claim_data.check_amount), insurance policies that started (customers.policy_start_date) before the company opened, and insurance policies that started (customers.policy_start_date) before a customer's birth date (customers.birthdate). Several other examples of unreasonable data that students often identified were negative values (claim_data.approved_amount and claim_data.check_amount), check amounts not equal to the approved amount (claim_data.check_amount and claim_data.approved_amount), and claims that appeared to be paid (claim_data.check_amount or claim_data.check_date) but were not in the "paid" status (claim_data.claim_status).

There were some data issues seeded throughout the case data set that only a few students identified. For example, only a few students detected dates out of sequence (claim_data.event_date, claim_data.claim_submission_date, claim_data.claim_decision_date, claim_data.check_date) or unusual house parts (claim_data.part_of_house). The least frequently detected

[17] We thank an anonymous reviewer for this suggestion.
[18] We thank an anonymous referee for suggesting this resource.

errors were duplicate customer names (customers.name),[19] customer IDs (customers.id), and claim IDs (claim_data.claim_id)[20], as identifying these errors required students to use a HAVING statement or to develop a subquery. In summary, while students had difficulty identifying a few data errors, they were able to detect many of the errors seeded throughout the data.

Many students found that responding to the two specific requests made by Mr. Serious were the most challenging part of the case (i.e., identifying customers whose total claim check payments (claim_data.check_amount) for the year exceeded the limits of their insurance policies (customers.house_value * coverage_limits.coverage_limit) and identifying claims for which the amount paid (claim_data.check_amount) is outside of the reasonable range estimated by the COO (between (customers.house_value * reasonable_ranges.bottom and customers.house_value * reasonable_ranges.top)). A few techniques appeared to underly students' challenges with these queries (joining data across multiple tables, creating calculated fields, and summing data by group).

To assist students struggling with these two queries, instructors found it useful to walk students through in-class exercises to develop queries that address a portion of the total requirement. For instance, instructors worked with students to develop a query that totals customers' claim check payments (claim_data.check_amount) and uses a HAVING statement to show which customers' total payments exceed a fixed amount (e.g., $500,000). Instructors next worked with students in class to develop a query calculating each customers' policy coverage limit (customers.house_value * coverage_limit.coverage_limit). On their own, then, students only needed to work through combining these two in-class queries. Similarly, instructors used in-class exercises to walk students through calculating the reasonable range of costs (between customers.house_value * reasonable_ranges.bottom and customers.house_value * reasonable_ranges.top) for a given customer's insurance claim Outside of class, then, students needed to compare the actual payments made by RHI (claim_data.check_amount) to the reasonable range (between customers.house_value * reasonable_ranges.bottom and customers.house_value * reasonable_ranges.top). Developing these example queries in class showed students that complex queries can be broken down into smaller, more manageable components. Showing students how to break complex problems down into smaller components helped students avoid frustration with the project. Given this assistance, most students were able to complete these two requirements effectively.

Acknowledgements

The authors wish to thank Natalie Churyk, Pamela Schmidt, and two anonymous reviewers for very detailed feedback that improved the quality of this case. We would also like to thank workshop participants at the University of Alabama ("Detecting Dirty Data" workshop, May 21, 2018, The University of Alabama, Tuscaloosa, AL) and at the 2018 American Accounting Association Intensive Data and Analytics Summer Workshop for Accounting Courses and Programs (Session 2.05 - "Detecting Dirty Data using SQL", June 5, 2018, Orlando, FL) for providing feedback on this case. Finally, we thank Gary Taylor and Anne Wu for their assistance on the case.

Appendix A:. Instructor resources

All of the following files are available from the authors upon request:

Student case files:

1) Case data in Microsoft Access, Microsoft Excel, or SQLite format (for use with TablePlus or SQL OnLine IDE). Alternatively, our case data is available using DB Fiddle at https://www.db-fiddle.com/f/6HxGYiyR5Ejyisr1vn5ujb/0.
2) Case overview and project requirements in PDF format.

Instructor resources:

1) Microsoft Powerpoint presentation that (a) introduces the Rigorous House Insurance data sets, (b) provides a hands-on introduction to SQL queries, and (c) introduces the eight-step process for detecting dirty data and explains why this process is a prerequisite for successful data analytics.
2) A PDF of the Powerpoint presention for instructors who are unable to access the actual Powerpoint (.pptx) file.
3) Instructor Key that provides a list of all the known dirty data issues seeded throughout the data.

[19] We acknowledge that two customers could have the same first and last names and yet be two distinct individuals. A query that simply required first and last name groups to have a count exceeding one could flag false positives. Students could address this possibility by also requiring that the group have a count of distinct birthdates exceeding one. We thank an anonymous referee for clarifying this point.

[20] In an effectively managed database, customer ID and claim ID would be designated as primary keys for the customer and claim tables, respectively. The database management system would then require unique values for these primary key fields – an information system preventative internal control preventing duplicate values. However, databases are not always effectively managed and primary keys are not always designated. In such an event, duplicate values can occur within fields which would intuitively seem to be primary keys. One of the authors encountered such a database in his professional experience.

4) Case Data Key (Microsoft Access file) that contains sample queries that can be used to review SQL in class. The Case Data Key also provides queries that satisfy the case requirements in part 2 of the case deliverable.
5) A well formatted entity relationship diagram.
6) Sample student submissions that include both the required memo (part 1 of the required deliverable) and the required queries (part 2 of the required deliverable).

References

AACSB International. 2018. "2018 eligibility procedures and accreditation standards for accounting acceditation." Available at: https://www.aacsb.edu/-/media/aacsb/docs/accreditation/accounting/standards-and-tables/2018-accounting-standards.ashx?la=en&hash=8DCDA6CE3B0CEF6AB82D39CBF53995DA96111196.

AICPA. 2020. "2017 Trust Services Criteria for security, availability, processing integrity, confidentiality, and privacy." Available at: https://www.aicpa.org/content/dam/aicpa/interestareas/frc/assuranceadvisoryservices/downloadabledocuments/trust-services-criteria.pdf.

Anderson, L. W., Krathwohl, D. R., & Bloom, B. S. (2001). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives* (Complete ed). New York: Longman.

Angelo, B., Ayres, D., & Stanfield, J. (2018). Power from the ground up: Using data analytics in capital budgeting. *Journal of Accounting Education, 42*, 27–39.

Borthick, A. F., Schneider, G., & Viscelli, T. (2016). Analyzing data for making: Integrating spreadsheet modeling and database querying. *Issues in Accounting Education, 32*(1), 59–66.

Cunningham, L., & Stein, S. (2018). *Using visualization software in the audit of revenue transactions to identify anomalies*. Issues in Accounting Education In-Press.

DaBruzzo, R., Dannenfelser, T., & DeRocco, D. (2013). Tax portals and dynamic data analytics: the new view for management and control.

Dzuranin, A., Jones, J., & Olvera, R. (2018). Infusing data analytics into the accounting curriculum: A framework and insights from faculty. *Journal of Accounting Education, 43*, 24–39.

Enget, K., Saucedo, G., & Wright, N. (2017). Mystery Inc: A big data case. *Journal of Accounting Education, 38*, 9–22.

Fay, R., & Negangard, E. (2017). Manual journal entry testing: Data analytics and the risk of fraud. *Journal of Accounting Education, 38*, 37–49.

Haselkorn, E. (2015). New Experian data quality research shows inaccurate data preventing desired customer insight Available at: http://www.experian.com/blogs/news/2015/01/29/data-quality-research-study/.

Igou, A., & Coe, M. (2016). Vistabeans coffee shop data analytics teaching case. *Journal of Accounting Education., 36*, 75–86.

ISACA (2018). *COBIT 2019 framework: Governance and management objectives* (p. 266). IL: Schaumburg.

Janvrin, D., Raschke, R., & Dilla, W. (2014). Making sense of complex data using interactive data visualization. *Journal of Accounting Education, 32*(4), 31–48.

Kokina, J., Pachamanova, D., & Corbett, A. (2017). The role of data visualization and analytics in performance management: Guiding entrepreneurial growth decisions. *Journal of Accounting Education, 38*, 50–62.

Loraas, T., & Searcy, D. (2010). using queries to automate journal entry tests: Agile machinery group, inc. *Issues in Accounting Education, 25*(1), 155–174.

McCabe, S. (2018). KPMG announces 2018 'Data and Analytics' master's program. Accounting Today. Available at: https://www.accountingtoday.com/news/kpmg-announces-2018-data-and-analytics-masters-program.

Press, G. (2016). Cleaning big data: most time-consuming, least enjoyable data science task, survey says. Forbes. Available at: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/.

PricewaterhouseCoopers (PwC). (2015). Data driven: What students need to succeed in a rapidly changing business world. New York, NY: PwC.

Romney, M.B. & Steinbart, P.J., (2018). Chapter 10: Processing integrity and availability controls, in: Accounting information systems (14th ed.). Pearson Education Limited, Harlow, England, 322-347.

Schneider, G., Dai, J., Janvrin, D., Ajayi, K., & Raschke, R. (2015). Infer, predict, and assure: Accounting opportunities in data analytics. *Accounting Horizons, 29*(3), 719–742.

Sherman, R. (2015). Business intelligence guidebook: From data integration to analytics. Morgan Kaufmann. ISBN 978-0-12-411461-6.

Sellers, R. D. (2017). SQL: An introduction to writing database queries. *Working paper*. Kent State University.

Vassiliadis, P. (2009). A survey of extract-transform-load technology. *International Journal of Data Warehousing and Mining, 5*(3), 1–27.

Zanzig, J., & Tsay, B. (2004). Hands-on training in relational database concepts. *Journal of Accounting Education, 22*(2), 131–152.

Zhang, J., Yang, X., & Appelbaum, D. (2015). Toward effective big data analysis in continuous auditing. *Accounting Horizons, 29*(2), 469–476.