User-Centred Database Query Interfaces: A Case Study

Jingxiu Cheng
HCI MSc Final Project Report 2021
UCL Interaction Centre, University College London
Supervisors: Enrico Costanza, Jon Geraghty, Rosie Prior

ABSTRACT

Database query interfaces are graphical user interfaces that allow users to interact with a database without using a database query language (e.g. SQL). However, previous work has not examined how users interact with a database query interface in a real-world setting, nor explored how database query interfaces should be designed for users who need to interact frequently with large databases. As such, this project had two aims: (1) investigate how employees interact with a database query interface in their company and the challenges they might face, and (2) design a user-centred database query interface that supports users who engage in regular querying, exploration and analysis of data from large databases. To achieve these aims, a database query interface at a data science company (Dunnhumby) that queries customer transaction data from the fast-moving consumer goods (FMCG) industry was used as a case study. Firstly, after conducting semi-structured interviews and usability tests with 15 employees at Dunnhumby, the database query interface was found to have a variety of use cases, to be useful in facilitating data exploration, and to be adopted only when users were confident in the interface. However, the database query interface was challenging to use due to poor visual design, an unintuitive user flow, poor visibility of system status, lack of support for browsing, unclear terminology, inefficient interactions and a lack of output flexibility. Secondly, by following an iterative and user-centred design process, this project came up with a novel design for a database query interface that received positive feedback during an evaluation with 6 participants. The empirical findings and design ideas from this project will provide designers and developers with a better understanding of user requirements and assist them in creating learnable and usable database query interfaces.

MSC HCI FINAL PROJECT REPORT

Project report submitted in part fulfilment of the requirements for the degree of Master of Science (Human-Computer Interaction) in the Faculty of Brain Sciences, University College London, 2021.

Author Keywords

Database query interface, visual query builder, data science, data exploration, data visualisation, user-centred design.

ACM Classification Keywords

•Human-centered computing → Empirical studies in HCI
•Human-centered computing → User interface design

MSc Contribution Type

Empirical, Design

1. INTRODUCTION

Although data science was traditionally assumed to follow an objective and unambiguous process, there has been a growing emphasis on the "humans" involved in data science work and the active role they play in shaping the analysis of data [24, 25, 31, 33]. Additionally, with the rise of big data [19], a growing number of non-data scientists – people who do not possess expertise in data science or related programming languages – are assuming roles that require them to interact with data [13]. Thus, there is a need to adopt a human-centred approach toward data science and examine how data analysis tools can be designed to better support both data scientists and non-data scientists at exploring, analysing and visualising data.

In particular, database query interfaces are commonly used to facilitate data analysis, especially by users who are inexperienced in data science [1, 6]. Database query interfaces refer to graphical user interfaces that enable users to interact with a database without requiring them to be familiar with a database querying language (e.g. SQL), and are designed to help users retrieve data from a database more easily and efficiently [5, 7]. The primary function of database query interfaces is to assist users in querying from a database, but database query interfaces also usually possess in-built tools for supporting users in running deeper analyses and generating visualisations [26].

Previous work on database query interfaces have studied the challenges users faced when using database systems [14] and proposed novel design ideas for database query interfaces [16, 34]. However, previous studies have not examined how users interact with database query interfaces in real-world settings, nor explored how database query interfaces should be designed to support users who need to interact frequently with data from large databases.

As such, the current project had two aims. Firstly, this project aimed to conduct an in-depth examination into how employees interact with a database query interface in their company and the challenges they might face when using the interface. Secondly, this project aimed to design a user-centred database query interface that supports users who engage in regular querying, exploration and analysis of data from large databases. To achieve these aims, this project was done in collaboration with Dunnhumby, a global data science company that works with retailers and brands in the fast-moving consumer goods (FMCG) industry. At the time of this project, Dunnhumby had recently introduced a database query interface that allowed employees in the company to interact regularly with large databases that contain the customer transaction data of their grocery retailer clients. Therefore, this particular database query interface at Dunnhumby was used as a case study for detailed investigation into this topic.

In the following sections, I will first review related work on human-centred data science, data analysis by non-data scientists and database query interfaces. Secondly, I will report on the semi-structured interviews and usability tests that were conducted with employees of Dunnhumby to explore their experiences with the database query interface. Thirdly, I will present the results of the user research, highlighting how employees in Dunnhumby interacted with the database query interface and the challenges they faced when using the interface. Fourth, I will elaborate on the iterative and user-centred process that I followed to design a database query interface that supported users in querying, exploring and analysing data from large databases. Lastly, I will discuss the implications of the findings in relation to the project aims and previous work, point out limitations of the project, and outline potential avenues for future work.

2. LITERATURE REVIEW

Human-Centred Data Science

Data science is often assumed to follow a passive and unambiguous process in which decisions are derived from objective and unbiased data [8]. However, this view neglects the role of the "human" in data science. In fact, humans play an active role in shaping the exploration and analysis of data, and use their subjective perspectives when interpreting and generating insights from the data [11]. Thus, we need to adopt a human-centred approach to data science and place a greater emphasis on the people who engage in data science work.

Recently, several studies have shifted toward this human-centric approach and focused on investigating the existing practices of data scientists when they interact with data. For example, Muller et al. conducted several interviews with data science workers using a grounded theory approach, and found that practitioners were actively involved in five main data science activities: (1) discovering data, (2) capturing data (e.g. data integration, data selection, data substitution), (3) curating data (e.g. data cleaning, data aligning), (4) designing data (e.g. imputing missing data, validating data) and (5) creating data (e.g. simulating data, interpreting data) [24]. In another study, Passi and Jackson conducted ethnographic fieldwork with a corporate data science team and found that data scientists face four main challenges in their work: (1) ambiguous numbers, (2) counterintuitive knowledge (3) untrustworthy data, and (4) incomprehensible models [25]. In addition, other studies have examined how data science practitioners collaborate with one another [31] or with artificial intelligence and machine learning technology [33]. Together, these studies provide us with a better understanding of the workflows of data scientists and the barriers they face in their work, which can inform the design of interventions and tools that assist practitioners in their data science work.

Additionally, other researchers have also adopted a human-centred approach and proposed novel ideas to support data scientists at data analysis. One example is progressive visual analytics (PVA), which has recently emerged as a method for facilitating data exploration [10]. Before PVA was introduced, the user had to follow a traditional analytics workflow, which involved them having to wait for an analytic to complete before they could inspect the results and re-launch the computation with adjusted parameters. Due to attention and memory constraints of human users, this process can disrupt the analyst's thought process and slow down the speed of analysis. To address these cognitive limitations, researchers came up with PVA, which allowed users to analyse partial results during the computation without having to wait for the analytic to fully complete [2]. Importantly, PVA was found to speed up analyses, increase engagement, and facilitate insight generation and hypothesis testing for a range of users including clinical researchers [28] and financial analysts [30]. Additionally, there is continued work toward creating user-centred PVA systems that cater to a range of target users who have different goals, tasks, focuses and biases [21]. Thus, PVA is a successful example of how a human-centred approach can facilitate the design of tools and solutions that support human users in data analysis.

Data Analysis by Non-Data Scientists

Although data science used to be restricted to professional data scientists and analysts, the rise of big data has resulted in an exponential increase in the volume of data science work [19]. As a result, a growing number of non-data scientists are taking up roles that require them to explore, analyse and visualise data [13]. These non-data scientists – also termed "information novices" [12, 18, 20] or "data enthusiasts" [23] – are defined as users who do not have formal training in data science and do not possess expertise in programming languages related to data science (e.g. SQL, Python, R). This new category of users typically carry out their data exploration, analyses and visualisation on business intelligence (BI) and data visualisation tools that use a graphical user interface (GUI) or a Windows, Icons, Menus, Pointers (WIMP) interface [13]. Examples of these tools include Tableau [29], Microsoft's Power BI [22] and IBM's Many Eyes [3].

As the number of non-data scientists that work with data continues to increase, multiple studies have been conducted to investigate the barriers that non-data scientists face when interacting with data. For example, Grammel et al. conducted a laboratory study in which non-data scientists were tasked to analyse sales data using commercial visualisation software [12]. Firstly, they found that the process of visualisation construction comprised of three primary activities: selecting the data attribute, selecting the visual template and specifying the visual mappings. Secondly, they identified the main barriers that non-data scientists face when exploring data; namely, non-data scientists found it challenging to translate abstract questions or goals into specific data attributes, and encountered problems in designing visual mappings and interpreting unfamiliar visualisations. Non-data scientists also tended to omit parts of visualisation specifications, and preferred to stick to familiar visualisation choices (e.g. bar charts). Based on these findings, they came up with a couple of design guidelines for data exploration tools. In particular, they proposed that these tools should provide search facilities to retrieve data, suggest possible visualisations, support iterative specification, assist users in recovering from partial specification errors, and provide explanations to support learning.

Similarly, to identify the roadblocks that non-data scientists face during data analysis, Kwon and colleagues conducted a qualitative experiment in which non-expert participants explored data using a visual analytics system [18]. Firstly, they found that non-data scientists often failed to choose the appropriate view to answer their questions. Secondly, they found that non-data scientists often failed to execute the appropriate interactions to display the information they needed to achieve their goal. Thirdly, non-data scientists sometimes faced difficulty reading and interpreting the visualisations. Fourth, non-data scientists' expectations sometimes did not match the functionality of the tool. Based on these findings, Kwon et al. proposed design recommendations to tackle each of these roadblocks, such as providing a default view, improving button and menu labels, adding helpful tooltips and guiding the user towards the appropriate mental models.

In addition, Morton et al. presented a vision for data analysis systems and highlighted the challenges of current tools in achieving this vision [23]. More specifically, they argued that existing visual analytic tools need to be more capable at data cleaning and data integration, and should recommend datasets that may be relevant to a user's primary task. Importantly, they argued that all these capabilities need to be seamlessly integrated into a single system that supports users in iterative data exploration.

While the aforementioned studies examined the potential challenges faced by non-data scientists when interacting with data, other studies have designed tools to support non-data scientists at analysing data. For instance, Elias and Bezerianos designed Exploration Views, a system that enables non-data scientists to easily create and customise business intelligence dashboards [9]. Similarly, TouchPivot – a novel interface that leverages WIMP interfaces and simple pen and touch interactions – was designed to support non-data scientists at data exploration on tablet devices [17]. Additionally, Yalcin et al. designed Keshif, a visualisation tool that automatically condenses data attributes into summaries thereby reducing unnecessary interactions and increasing exploration speed [32]. Apart from these examples, many interfaces and tools are continually being developed to support non-data scientists at performing data exploration, analysis and visualisation.

Database Query Interfaces

One type of tool that was designed to support non-data scientists at data analysis is the database query interface. Within a company, data is typically stored in a database system, and users would need to use a database querying language (e.g. SQL) to access and interact with the data. However, console-based querying is difficult to understand and is not accessible to users who do not possess expertise in a database querying language [14]. As such, database query interfaces, also sometimes called visual query builders or graphical database interfaces, were developed to support users – especially those who are inexperienced in data science – at querying from a database [26]. Database query interfaces refer to graphical user interfaces that make the querying process highly visual and straightforward, and have received attention both in the academic literature [1, 6] and in applied settings [5, 7].

However, although database query interfaces are meant to be user-friendly and easy to learn, many database query interfaces today are still highly complex and difficult to use [15]. Jagadish and colleagues conducted a study to examine how users interacted with query interfaces to access a database, and outlined several pain points that affected the usability of database systems [14]. In particular, one challenge of database systems was finding the right balance between simplicity and functionality; the database system should not have too many options so novice users can learn to use it, but needs to possess enough functionality so expert users can carry out more complex queries. Another limitation was that database systems did not always align with the mental models of users; as a result, users frequently experienced confusion when querying the databases or were surprised at unexpected results. One other barrier of database systems was that they did not follow the property of "what you see is what you get" (WYSIWYG), and were overly dependent on the user to accurately predict the outcome of their queries. Although the challenges identified by Jagadish et al. were for database systems in general, many of these limitations are also relevant to database query interfaces.

In an attempt to make database query interfaces more usable, several studies have adopted a user-centred approach toward designing these interfaces. For instance, Zhang et al. designed VISAGE, a database query interface that aimed to assist clinical investigators in directly assessing a clinical research database [34]. During their user research, they identified that users wanted the interface to provide them with instantaneous feedback so they could iteratively refine their query, and wanted the ability to save and re-use previous queries. Therefore, one feature of VISAGE was a Query Manager that allows users to easily construct their queries and save previous queries for future use. Another feature of VISAGE was a Query Explorer, in which users could get quick descriptive statistics from the results (e.g. mean, standard deviation) and generate simple graphs (e.g. pie charts, histograms). In a preliminary evaluation, VISAGE was found to reduce the number of clicks and time taken to formulate a query.

In another study, Jiang and colleagues followed a user-centred process to design GestureQuery, a database query interface for multitouch tablet devices that allowed users to use gestures to directly interact with the data [16]. Importantly, their interface followed the WYSIWYG property, enabling the user to interact with the data in a visual manner and providing immediate feedback to support the user's train of thought. In a preliminary evaluation, GestureQuery was found to be quicker than console-based querying and performed at a comparable speed to other visual query builders. Overall, the novel designs of VISAGE [34] and GestureQuery [16] have revealed features and concepts that can be useful for improving the learnability and usability of database query interfaces.

Rationale of Current Project

Previous research has highlighted various challenges that affect the learnability and usability of database systems [14] and other data analysis tools [12, 18, 23], especially for non-data scientists and other non-technical users. However, none of these previous studies have specifically focused on database query interfaces and investigated how users interact with database query interfaces in a real-world context. Hence, the first aim of the present project was to examine how employees interact with a database query interface in their company and the challenges they might face when using the interface. To achieve this aim, a specific database query interface at a data science company (Dunnhumby) was used as a case study for this project.

Additionally, even though some studies have adopted a human-centred approach toward the design of database query interfaces [16, 34], there is limited literature on how database query interfaces can be designed effectively. Furthermore, previous studies have focused on database query interfaces that would only be used infrequently and for querying relatively small databases (e.g. clinical research database [34] and digital media database [16]). Thus, the second aim of the current project was to design a user-centred database query interface that supports users who engage in regular querying, exploration and analysis of data from large databases. In particular, the novel database query interface was designed for querying customer transaction data in the competitive fast-moving consumer goods (FMCG) industry, since huge databases are involved (i.e. millions of rows) and users are expected to formulate large volumes of queries on a regular basis.

3. METHOD

The first aim of this project was to examine how employees in a company interact with a database query interface and the challenges they might face. To achieve this aim, user research was conducted with employees of Dunnhumby to explore their experiences with a database query interface that was used internally within the company.

Participants

The study was conducted on 15 participants who were employees of Dunnhumby, a global data science company that works with retailers and brands in the fast-moving consumer goods (FMCG) industry. Participants were recruited through convenience sampling; study invitations were sent out through email to employees who were thought by the industry supervisor of the project to be potential or existing users of the existing database query interface. Participants did not receive any compensation, and ethical approval was granted by the UCL Interaction Centre (UCLIC) Research Department's Ethics Committee (Project ID Number: UCLIC/1617/017/Staff Costanza/Nowacka/Yang) and by Dunnhumby's internal data governance board.

Participants worked in a variety of roles within the company (e.g. Client Leads, Media Planners, Data Scientists, Team Managers) and made use of customer transaction data in different ways. Participants had varying levels of data science expertise; some participants had less than two years of experience in working with customer data and had no programming expertise, whereas other participants had over 15 years of experience in working with customer data and were proficient in programming languages related to data science (e.g. SQL, Python). 10 of the participants were regular users of the existing database query interface (i.e. used it at least once a month), whereas 5 of the participants rarely or never used the interface.

Materials

During the study, participants performed tasks using a database query interface that was introduced into Dunnhumby a few years ago (Figure 1). The tool had a simple forms-based user interface that allowed users to interact regularly with large databases containing the customer transaction data of Dunnhumby's grocery retailer clients. The tool utilised real-time querying (as compared to batch querying), which meant that users could receive their query results quickly (i.e. within seconds) rather than having to wait for hours or days for a batch job to finish [27]. Importantly, the database query interface enabled users to retrieve customer metrics (e.g. customers, spend, visits) at the macro level (e.g. across categories, stores and

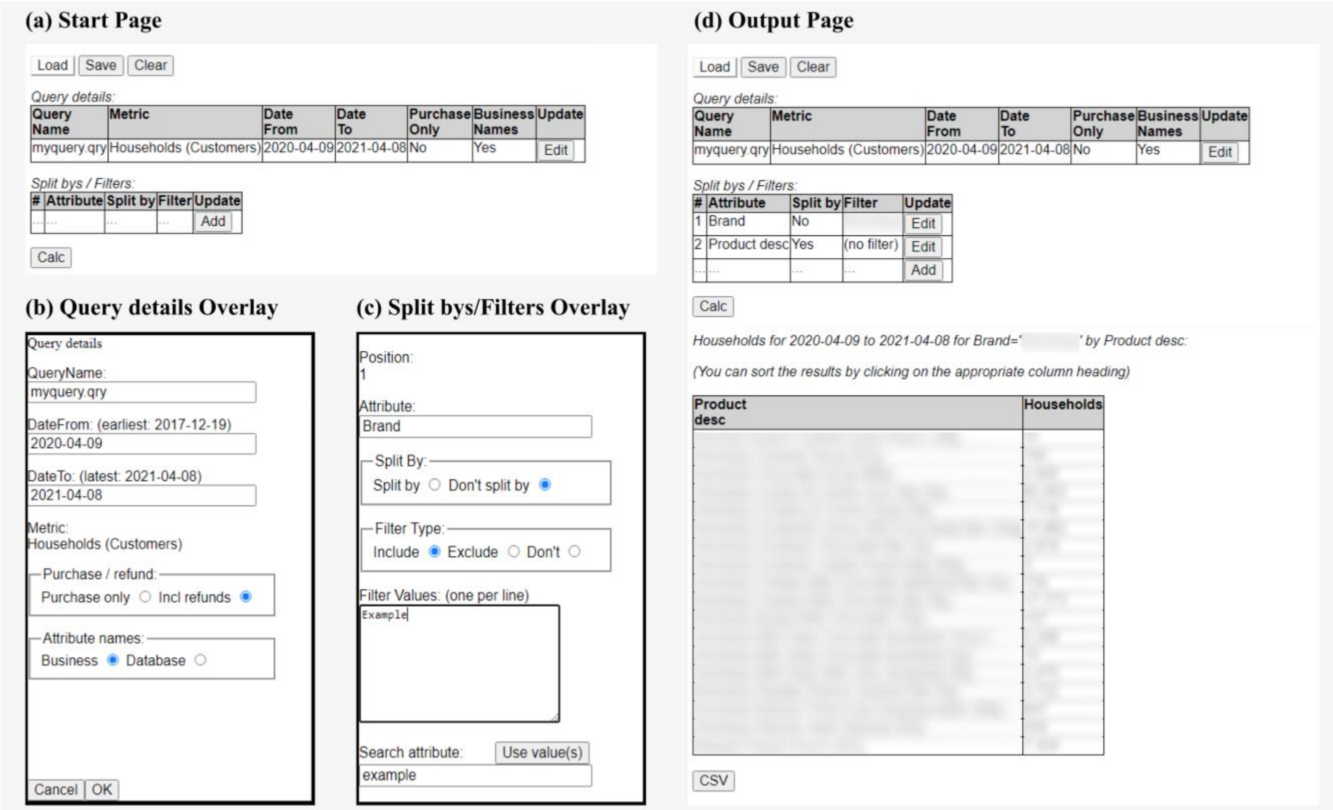

Figure 1. The database query interface at Dunnhumby. Figure 1a shows the start page. Figure 1b shows the overlay for the query details that appears when the 'Edit' button (start page) is clicked. Figure 1c shows the overlay for the split bys and filters that appears when the 'Add' button (start page) is clicked. Figure 1d shows the output that appears when the 'Calc' button is clicked.

shopping channels) and the micro level (e.g. specific products, customers segmentations and promotions).

The user flow of the database query interface follows three main steps. From the start page (Figure 1a), participants first enter their query details, which include the customer metric and dates they were interested in (Figure 1b). Secondly, participants add attributes which they can use to filter or group the data (Figure 1c). Thirdly, after participants finish formulating their query, they press a "Calculate" button to get the results (Figure 1d). The output is then presented in a table which can be exported as a CSV (comma-separated values) file.

Design & Procedure

The study was conducted remotely using video conferencing software (Microsoft Teams) and lasted approximately 45 minutes. Participants were presented with an information sheet and a consent form to read and complete in advance of their study session. A topic guide was also developed beforehand to facilitate the semi-structured interviews and usability tests during the study.

In the first part of the study, a semi-structured interview was conducted to better understand the users and their current practices around interacting with the database query interface. In particular, participants were asked to describe their role at Dunnhumby, their experience in working with customer data and their expertise in programming languages or tools related to data science. They were also asked to elaborate on their goals and workflows when interacting with the database query interface, and the general barriers they faced when using the interface.

In the second part of the session, a usability test was conducted to identify the specific challenges users faced when using the database query interface. Participants were tasked to think about something they might be interested in finding out from the customer transaction data, and to use the interface to accomplish that task. Participants were reminded to think aloud throughout the usability test. After participants completed or gave up on the task, they were asked some follow-up questions. More specifically, they were probed to elaborate on the specific limitations of the interface and to provide suggestions on how they think the interface could be improved.

Analysis

Due to privacy reasons, only 6 of the participants had their study sessions video-recorded and transcribed. During the transcription, personally identifiable information or sensitive data were made anonymous or removed. An inductive (bottom-up) analysis was then conducted on the transcripts using a thematic analysis approach [4]. Firstly, the transcripts were read several times to gain familiarity with the data. Secondly, relevant features of the data were labelled using codes that were developed inductively. Thirdly, similar codes with coherent patterns were combined to form themes. Fourth, these themes were reviewed to ensure that they meaningful related to the original codes and to each other. Finally, informative names and definitions were given to each of the themes.

For the remaining 9 participants, anonymous notes were taken during the study session. These notes were also analysed following an inductive thematic analysis approach; similar notes that related to each other were grouped together in an affinity diagram to form themes.

Since the analyses on the transcripts and notes resulted in similar codes and themes, the results are reported together in the following section.

4. RESULTS

The findings from the user research are reported in two main sections. The first section describes how employees of Dunnhumby interacted with the database query interface. The second section highlights the challenges faced by users due to the design limitations of the database query interface.

Interactions with the Database Query Interface

This section outlines three themes that describe the use cases of the database query interface, its usefulness for data scientists and non-data scientists, and the factors that affected user adoption of the database query interface.

Theme 1: Database Query Interfaces have Many Use Cases

The database query interface at Dunnhumby was used for a wide range of tasks, even though participants had a variety of different roles and belonged to many different teams within the company. Employees in the client team – who are responsible for communicating insights to clients and managing relationships with clients – mainly used the database query interface to obtain key performance indicators (KPIs) for their presentations and to respond to urgent ad-hoc requests from clients. Employees in the media team – who are responsible for planning and executing media campaigns – typically used the database query interface to identify target products to focus on and to monitor the performance of their campaigns. Interestingly, even data scientists – who wrote their own code in a database query language to retrieve data – frequently used the database query interface to explore the data or to check that their code was running correctly.

“I use [the database query interface] to get the basic statistics before a meeting, so I can present to the client and give them a better understanding of what the scope of the project is.” (Client Manager, P2)

“I use [the database query interface] for sense checking certain numbers and to make sure the output from my code looks right.” (Data Scientist, P7)

Additionally, the database query interface was used by participants throughout the different stages of their workflow. At the beginning of a project, participants used the data query interface to identify aspects of the data to focus on and to determine the scope of their projects. In the middle of a project, participants used the data query interface to check whether their projects were running smoothly and to spot any early errors. Toward the end of a project, participants used the data query interface to evaluate the success of their projects and to get data to support their presentations and reports.

“We start analysing data [using the database query interface] even from the beginning before running the campaign to see if it’s feasible or not. If it is feasible I check the data throughout the campaign and use it to create the last reports at the end.” (Media Planner, P4)

“At the early concept stage, I used [the database query interface] to shape the project brief. At the later stages of the project, I conduct validation checks. At the final stage, I am usually preparing presentations, so I use [the database query interface] to get numbers for my figures which is used to support my narrative.” (Senior Data Scientist, P12)

Theme 2: Database Query Interfaces Facilitate Exploration

The database query interface was found to support users at exploring data. Firstly, the database query interface facilitated data exploration for non-data scientists by making rawer forms of the data more accessible to them. Before the implementation of this database query interface, non-technical users were unable to interact directly with the database, even though majority of their work at the company involves the use of customer transaction data. Instead, they had to ask a data scientist within the company to run queries or analyses for them. As such, it was challenging for non-data scientists to explore the data independently or generate their own insights. Thus, by increasing the accessibility of raw data to non-technical users, the database query interface facilitated data exploration for non-data scientists.

“I love [the database query interface] because it empowers me with the ability to explore the data myself... it allows me to bring in my own perspectives and come up with my own insights from the data.” (Client Lead, P10)

“Having my own ability to access the data would mean that I can be a bit more self-serving and do something independently without needing to increase the workload for my team of analysts.” (Client Manager, P2)

Secondly, the database query interface supports data exploration by reducing disruptions to a user’s train of thought. With the database query interface, users could formulate their queries quickly and obtain the results within seconds. This allowed users to easily explore the data to find interesting aspects to focus on, and enabled users to iteratively refine their queries toward specific hypotheses. On the other hand, if a non-data scientist wanted to ask a data scientist to analyse the data for them, they would need to already have a pre-conceived notion of the data they were interested in, and would not be able to refine their research questions based on intermediate findings. In addition, the long periods which they would have to wait before hearing back from the data scientist could be disruptive to their thought process and workflow. Similarly, when data scientists are querying the database using code, it can be costly to make errors as they would need to spend a long time debugging and editing their code. With the database querying tool, data scientists were able to quickly explore the data for points of interest before running deeper analyses on them. Hence, by increasing the speed and ease in which one can analyse the data, the database query interface supports both non-data scientists and data scientists at data exploration.

"With [the database query interface], you can get quick metrics rather than having to wait… If you wanted to change something or look at it differently, waiting can break the momentum that you've got." (Client Lead, P3)

"[The database query interface] is fast and I have the freedom to make many mistakes… I can quickly get answers to new questions that I'm asking." (Client Lead, P8)

Theme 3: Adoption Depends on Confidence, not Expertise

Interestingly, adoption of the new database query interface was not dependent on the participants' level of expertise in data science. In particular, some of the participants who had many years of experience working with customer data and were proficient in programming languages related to data science (e.g. SQL, Python) were not able to successfully learn how to use the database query interface. In contrast, many participants who had little experience in data science managed to learn how to use the database query interface.

"I use [the database query interface] around 2 to 3 times a week … After getting over the initial learning curve, it is usable and complements my workflow." (Client Lead, P13)

"I don't use [the database query interface] because I am not sure if I can get what I want from using it." (Senior Data Scientist, P6)

Instead, successful adoption of the database query interface appeared to depend on the confidence of the user. Participants who did not use the database query interface found the interface confusing and intimidating, and did not believe that they could use the tool correctly to achieve their desired output. On the other hand, participants who ended up using the database query interface regularly were confident in using the tool and trusted the accuracy and reliability of the output.

"For me, I feel more confident in getting an expert to do it versus myself… I guess if I was more confident in using it I might use it more, but now I just keep second guessing myself." (Client Lead, P1)

"I use [the database query interface] often because I trust the output more than my own code… [The database query interface] is my truth." (Data Science Manager, P15)

Limitations of the Database Query Interface

This section outlines seven design limitations of the database query interface that affected its learnability and usability. Although the barriers highlighted in this section are specific to the database query interface at Dunnhumby, many of these limitations are also generally applicable to other existing database query interfaces.

Limitation 1: Poor Visual Design

A first limitation of the database query interface was its poor visual design. The current user interface (Figure 1) was not designed using a human-centred approach; rather, a simple user interface was quickly put together using a set of default widgets and interactions (i.e. jQuery UI library). As a result, the user interface did not apply any principles of visual hierarchy (e.g. size, colour, contrast, spacing), making it unclear how the interface could help the user to achieve their goals. As such, there was poor user satisfaction and a lack of confidence in the interface.

"The interface looks plain and bland." (Client Lead, P8)

"The interface does not appeal to me visually." (Senior Data Scientist, P12)

Limitation 2: Unintuitive User Flow

A second limitation was that the database query interface had an unintuitive user flow. Currently, users had to (1) select the metric they were interested in (e.g. number of customers) and add specific filters (e.g. date range), before (2) selecting the attributes (e.g. brands) that were either used as filters or to group the data (Figures 1b and 1c). Some participants found the separation of the two steps confusing, and other participants found it unnatural that adding filters and grouping the data by attributes were done at the same step. Thus, the user flow of the database query interface was not aligned with users' mental models, which resulted in confusion and frustration.

"Why is choosing the metric and splitting or filtering in two separate boxes?" (Data Scientist, P14)

"I find that it is more natural to filter the data first before I choose how I want to split the data into groups." (Data Science Manager, P15)

Limitation 3: No Visibility of System Status

A third limitation was the lack of visibility of system status when a user is formulating a query. As a user is building their query, they had no idea what the output was going to look like until the query had been run. If users made any errors, they would only notice it late in the process, when they were presented with an unexpected output (e.g. if they selected the wrong attributes to split their data by) or an error message (e.g. if they selected dates that are beyond the available date range). Hence, the database query interface had poor usability due to poor error prevention and visibility of system status.

"Looks like there's an error... I am not sure where I made a mistake." (Client Manager, P2)

"[After being presented an error message] It will be useful if unavailable dates are not shown in the first place." (Client Lead, P10)

Limitation 4: Does not Support Browsing

A fourth limitation was that the database query interface did not support browsing. There were many customer metrics which users could look at (e.g. household penetration, card penetration) and many attributes which users could use to filter or group the data (e.g. brand, time period). However, many of these useful attributes and metrics were not noticed because they were hidden at the bottom of a long dropdown list. Additionally, when users were filtering by specific values within an attribute (e.g. specific products), they needed to remember the names of those values because a list of available values were not provided to them. Thus, due to the poor discoverability of attributes and the heavy dependence on memory, the database query interface was limited in supporting users at formulating their queries.

"This is the first time I'm looking through this entire list [of metrics] and I realise that many of these are actually useful for me." (Client Lead, P13)

"I didn't realise that there were so many options for the attributes." (Client Lead, P3)

Limitation 5: Unclear Terminology

A fifth limitation of the database query interface was that many terms within the interface were unclear or confusing. For instance, many users did not understand the difference between metrics (i.e. database fields with numerical values, such as customers or spend) and attributes (i.e. database fields with text values such as product description or brand which users can use to filter or group the data). Additionally, many users did not understand the meaning of some of the more complex metrics or attributes. However, there was no explanation for the terms except in a separate 'Help' section that users tended to ignore. Due to the unclear terminology, many users felt unsure and lacked confidence when they were building their query.

"The metric names are confusing... I wish there was a glossary for the terms that is easily accessible." (Client Lead, P8)

"I am confused by the attribute names… I never bothered to try to find the answer in the 'Help' section because it was too troublesome." (Data Science Manager, P15)

Limitation 6: Inefficient Interactions

A sixth limitation was that many of the interactions within the database query interface were inefficient, leading to many sources of friction. For example, multiple clicks were required for user to shift the order of their split bys and filters or to delete unused attributes (Figure 1c). In addition, users could not run multiple metrics at once; instead, for each metric they wanted to examine, they needed to change the metric in the query builder and run the query again. Furthermore, while users often run queries that were similar to what they ran previously, the feature for saving queries was non-intuitive and many users did not even know that there was function that allowed them to save queries. Overall, the many inefficient interactions present in the database query interface affected the usability of the tool.

"I wish there was an easier way to delete filters and split-bys… it will also be nice to be able to run multiple metrics at once rather than having to run multiple queries." (Senior Data Scientist, P12)

"I tend to re-use what I've already got rather than rebuild a query. I will want to save certain queries, but I don't know if it's possible or how I can do it." (Client Lead, P9)

Limitation 7: Lack of Flexibility in Output

Lastly, a seventh limitation of the database query interface was the lack of flexibility given to the output (Figure 1d). Beyond sorting the results table by specific columns, there was no manipulation that could be done to the output unless the user went through the effort of exporting the file into a separate spreadsheet tool. Since many users utilised the database query interface simply for data exploration or for getting quick results, they wanted the ability to create pivot tables or view simple visualisations within the tool itself. Hence, the lack of flexibility in manipulating the output was a final limitation that affected user satisfaction.

"I would like some flexibility in the output, such as sorting or pivoting the table, so I don't have to copy and paste the results into Excel and manually manipulate it." (Client Lead, P10)

"It will be nice to be able to get some quick visualisations so I can see the results with a different lens when exploring the data." (Data Scientist, P7)

5. DESIGN PROCESS

The second aim of this project was to design a database query interface that supports users who engage in regular querying, exploration and analysis of data from large databases. To achieve this aim, an iterative and user-centred design process was followed. Firstly, personas and user requirements were developed based on findings from the user research. Secondly, design ideas were generated to address limitations of existing database query interfaces. Thirdly, the design ideas were combined into an interactive prototype and evaluated through usability testing.

Personas and User Requirements

Based on findings from the user research, two personas were created to describe the typical potential user and the typical regular user of the database query interface. The personas were split into these two groups because users belonging to each persona tended to have distinct profiles, motivations and barriers.

The first persona describes potential users of the database query interface (Figure 2). Although these users believed that the database query interface would be useful in supporting them to accomplish their tasks, they were never able to learn how to use it. They usually have client-facing roles or other non-technical roles and do not come from a data science background. They would like to use the tool because it would support their workflow, allow them to be more self-sufficient and help them to get the data they need more quickly. However, because of the limitations of the interface, they lack the confidence to use the database query interface independently and require support from their colleagues to run their queries. Overall, the main user requirements for potential users are that the database query interface is intuitive and easy to learn.

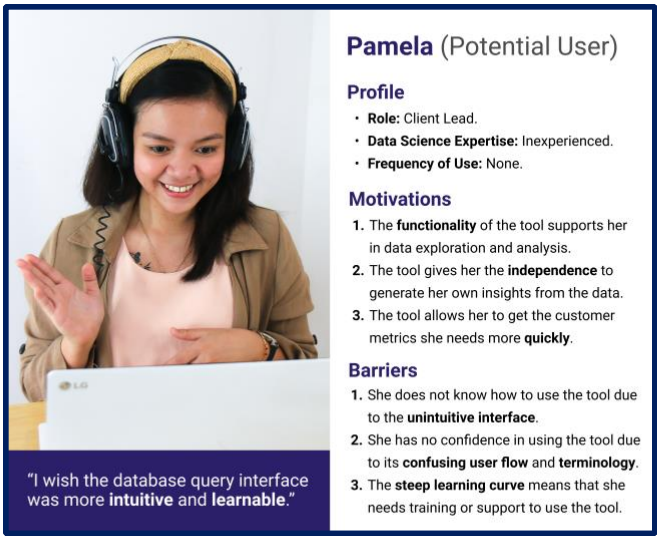

Figure 2. Persona for a potential user of the database query interface.

The second persona describes regular users of the database query interface (Figure 3). These users have managed to overcome the learning curve of the database query interface and now use the interface frequently. They usually have more technical roles and have a stronger background in data science. They enjoy using the tool because of its functionality, speed and reliability. However, they find that some aspects of the database query interface are inefficient and inflexible, and fail to discover many functions that might help them to achieve their goals. Overall, the main user requirements for regular users are that the database query interface is efficient and flexible to use.

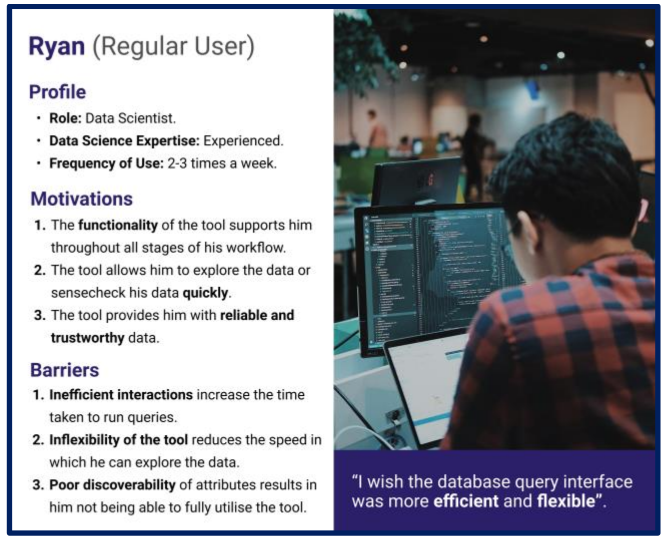

Figure 3. Persona for a regular user of the database query interface.

Design Ideas

After establishing the user requirements of each persona, design ideas were generated with the aim of enhancing the learnability and usability of a database query interface. Ideation first began with a diverging process, which resulted in the production of a large quantity of ideas. Ideation then proceeded toward a converging process, in which initial ideas were refined, combined or removed based on how effective they were in meeting the user requirements. At the end of the process, seven design ideas were generated. The ideas were first drawn as sketches (Figure 4) and were later combined into an interactive prototype (Figure 5) using a wireframing and prototyping tool (Figma). This section elaborates on how the design ideas addressed the limitations identified in the user research to create a database query interface that effectively supports users in the regular querying, exploration and analysis of data from large databases.

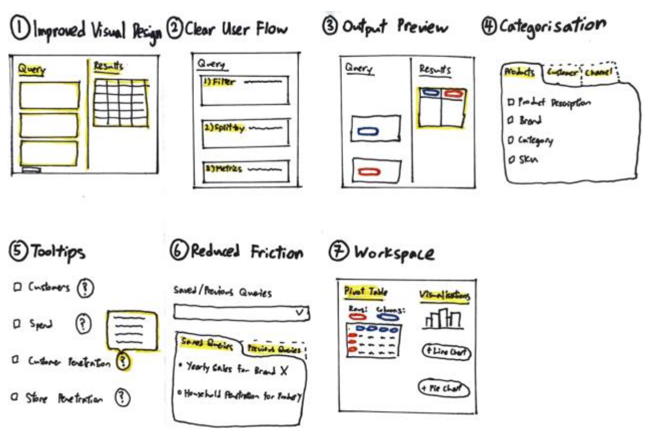

Figure 4. Sketches of the seven design ideas.

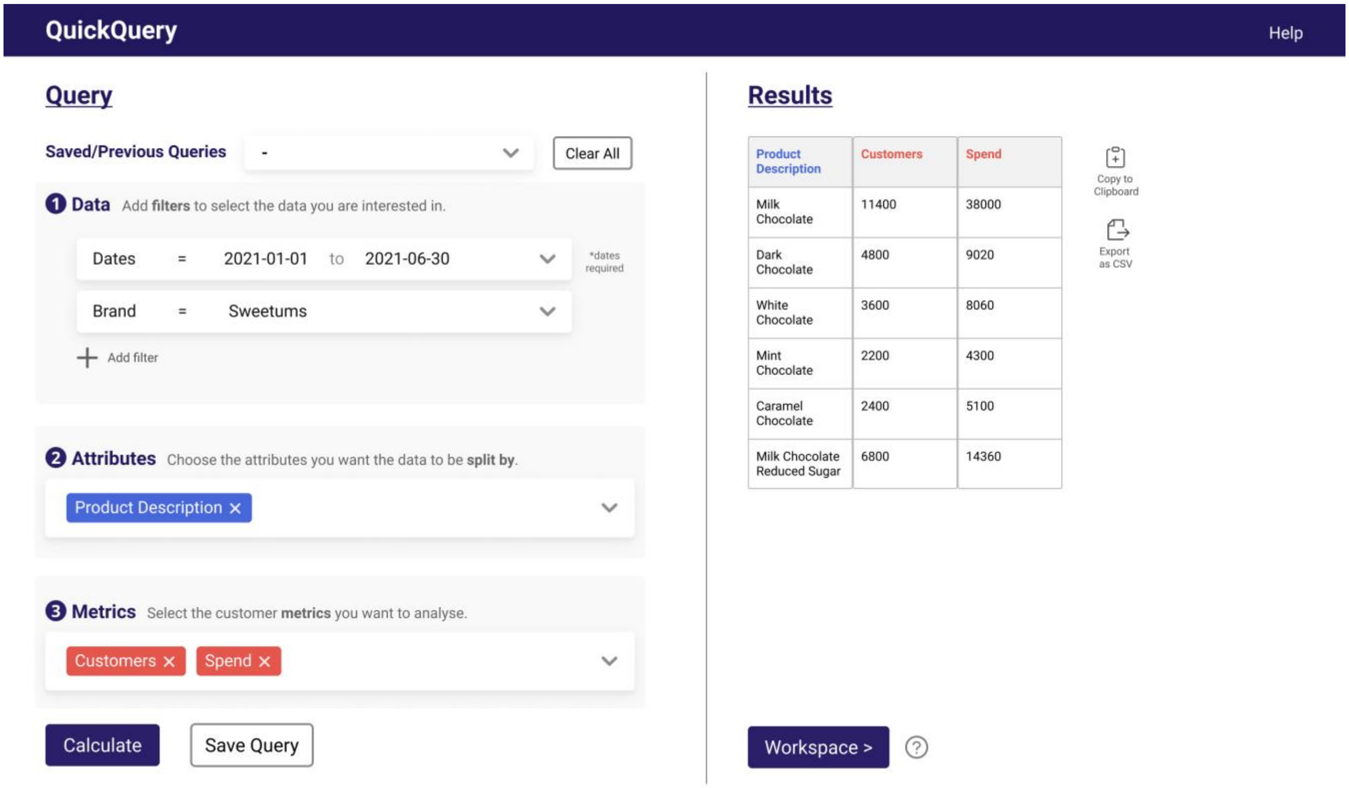

Figure 5. Interactive prototype showing the new design of the database query interface.

Idea 1: Clean Visual Design

The first idea was to create a clean visual design for the database query interface (Figure 5). In particular, the new interface applies principles of visual hierarchy by using colours, contrast and spacing to highlight the different sections. More colours and modern design elements were also added to make the interface more visually appealing to users. Together, these features aim to increase user satisfaction and instil confidence in users when they interact with the database query interface.

Idea 2: Clear and Intuitive User Flow

The second idea was to create an intuitive user flow for formulating queries. With the new interface, users will first (1) filter the data they are interested in, then (2) choose the attributes they want to group the data by, followed by (3) selecting the metrics they are interested in analysing. This user flow aims to provide a more intuitive querying process by aligning with the mental models of users. Additionally, the different steps are clearly demarcated and explained, so new users who are unfamiliar with the interface can follow a clear step-by-step guide to formulate their query. Importantly, familiar users still have the flexibility of building their query in the order that they prefer (e.g. completing steps 2 and 3 before step 1). Overall, these features aim to improve the learnability of the interface for new users, while providing existing users with the flexibility to follow their preferred way of building queries.

Idea 3: Output Preview

The third idea was to allow users to see a preview of their output as they are formulating their query. More specifically, as a user is constructing their query (on the left of the interface), a preview of their output table will simultaneously be shown (on the right of the interface) before the calculations are run (Figure 6). This feature provides users with immediate feedback and a "What You See Is What You Get" (WYSIWYG) interface, so they can better understand what they are doing and quickly adapt their query to fit their goals. Thus, this idea aims to communicate the visibility of system status to users and support them in error recognition and recovery.

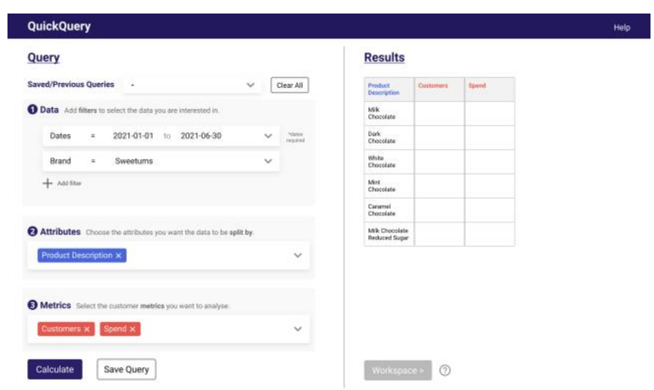

Figure 6. Output preview aims to assist users in recognising and recovering from errors.

Idea 4: Categorisation of Attributes and Metrics

The fourth idea was to group attributes and metrics into categories (Figure 7). This feature allows users to see what attributes and metrics are available to them at a glance, and enables them to easily discover similar attributes or metrics that might be useful to them. Hence, this feature aims to support users at discovering new ways of using the database query interface to make full use of its functionality.

Figure 7. Similar terms were grouped together to improve the discoverability of attributes and metrics.

Idea 5: Tooltips

The fifth idea was to add tooltips that contained expandable text beside terms that could potentially be confusing or unclear (Figure 8). The tooltips contain definitions of complex terms and links to specific reference lists (e.g. product hierarchy for the grocery store), providing just-in-time clarification for the user. Therefore, this idea aims to improve the learnability of a database query interface by making it more understandable to the user.

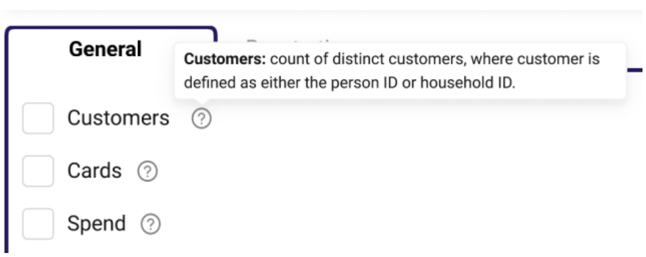

Figure 8. Tooltips aim to reduce confusion for the user.

Idea 6: Reduced Friction

The sixth idea was to add features to reduce the friction of building queries. For example, users are able to select multiple attributes to split the data by at once, or choose multiple metrics to calculate at once. In addition, metrics and attributes that are not available (e.g. dates outside the available date range) are no longer selectable to users. Moreover, users are given the ability to save queries that they run frequently or re-use queries that they have ran previously, so they do not have to manually build them each time (Figure 9). These features aim to prevent errors and reduce the number of clicks required by the user, in order to make the query building process as efficient and streamlined as possible.

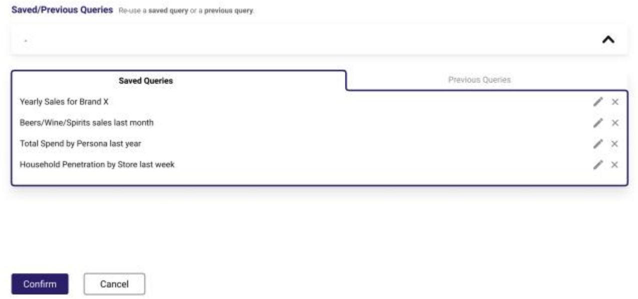

Figure 9. Saved/Previous Queries aim to increase the efficiency and usability of the database query interface.

Idea 7: Workspace

Finally, the seventh idea was to create a "workspace" where users could quickly manipulate the result of a query. After users have calculated their query, they can click a button that brings them to the workspace (Figure 10). In the workspace, users can use pivot tables to group, sort and aggregate their results in different ways. In addition, users can add simple visualisations for their results. If users wish to run more thorough analyses on the data, they can copy the results to their clipboard or export the CSV file into a separate spreadsheet or business intelligence tool. Together, these ideas aim to support users at rapid data exploration and insight generation.

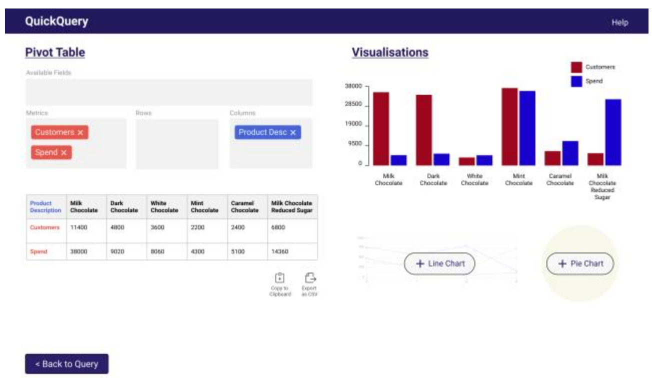

Figure 10. Workspace aims to facilitate data exploration and train-of-thought analyses.

Evaluation

Usability tests were conducted to evaluate whether the new database query interface was effective at supporting users who engage in regular querying, exploration and analysis of data from large databases.

Method

The evaluation was carried out with 6 employees of Dunnhumby, who agreed to a follow-up session after participating in the previous round of user research. 4 of the participants were regular users of Dunnhumby's existing database query interface tool (i.e. uses the tool at least once a month), whereas 2 of the participants rarely used the tool. Participants did not receive any compensation and ethical approval was granted by the UCL Interaction Centre (UCLIC) Research Department's Ethics Committee

(Project ID Number: UCLIC/1617/017/Staff Costanza/Nowacka/Yang) and by Dunnhumby's internal data governance board.

The evaluation was conducted remotely using video conferencing software (Microsoft Teams) and lasted around 30 minutes. Before the evaluation session, participants were presented with an information sheet and a consent form, and a topic guide was developed to facilitate the session. Additionally, a fictional scenario was created to serve as a user task for the usability test.

During the session, participants were asked to perform the user task using the interactive prototype (Figure 5). Participants were reminded to think aloud during the task. After they completed the task, participants were asked to browse through the various screens of the prototype and provide their feedback on the design ideas. Anonymous notes were recorded during the evaluation session and analysed through affinity mapping (Figure 11).

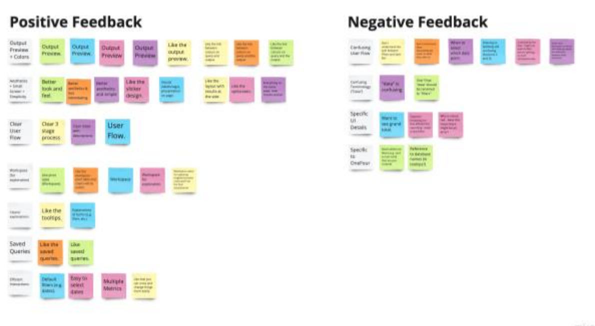

Figure 11. Affinity diagram in Miro. Insights were split into positive and negative feedback, and similar points were grouped together to form themes (white sticky notes).

Results

Overall, the overall design and the specific design ideas were well-received by participants. All of the participants liked the clean and aesthetic design (Idea 1), especially the compact layout and the modern and colourful visual elements. All of the participants also especially liked the output preview feature (Idea 3), and how they could easily tell which part of the query was linked to which part of the results based on the colours of the attributes. Many of the participants also felt that there was less friction when using the interface (Idea 6), since interactions were efficient and they could save and re-use previous queries. Additionally, many participants liked the workspace (Idea 7), but only for quick data exploration rather than for more complex analyses. Some participants also pointed out that that they liked the categorisation of attributes and metrics (Idea 4), and that the tooltips (Idea 5) helped them to understand some of the more complex terms.

In contrast, many of the participants were initially confused by the new user flow (Idea 3). In particular, they did not understand what filtering the data (step 1) and splitting the data by attributes (step 2) referred to, and why they were put into separate steps. However, this could be because the new user flow was different compared to the existing database query interface that users were familiar with. In fact, after playing around with the new design, some of the same participants felt that the new user flow was clearer and more intuitive. Nevertheless, to improve the design, the first two steps will subsequently be renamed to "Filter" (instead of "Data") and "Group-by" (instead of "Attributes") to make it clearer what each step refers to. When users are adding filters, a summary of their filters will also be shown in the output preview to give users feedback on how their actions are affecting the output.

In addition, participants also suggested some minor changes to the user interface. These include adding a search bar for attributes/metrics, changing the date format to make it more readable, and adding row and column totals in the results table. Nonetheless, apart from these specific suggestions for improvement, the new database query interface was generally thought to be learnable, usable and effective at supporting users in querying, exploring and analysing data.

6. DISCUSSION

Examining Interactions with a Database Query Interface

The first aim of this project was to examine how employees interact with a database query interface in their company and the challenges they face when using the interface. To achieve this aim, semi-structured interviews and usability tests were conducted with 15 employees of Dunnhumby to understand their experiences of interacting with a database query interface in their company. Firstly, the database query interface was found to have many use cases for both data scientists and non-data scientists, and was used throughout different stages of their workflow. Secondly, the database query interface was found to facilitate data exploration by giving non-data scientists access to the database and by reducing disruptions to users' flow of thought. Thirdly, user adoption of the database query interface was shown to be dependent on the confidence of the user rather than their level of expertise in data science. Lastly, users were found to face challenges in using the database query interface due to several design limitations, which included its (1) poor visual design, (2) unintuitive user flow, (3) poor visibility of system status, (4) lack of support for browsing, (5) unclear terminology, (6) inefficient interactions and (7) lack of output flexibility.

Interestingly, many of the results from this project were similar to findings from previous literature. For instance, this project found that the database query interface was used by a range of technical and non-technical users in the company for many different purposes. This finding suggests that despite its plain and non-intuitive user interface, the database query interface was simple enough for some non-data scientists to learn and had sufficient functionality that it was useful to both data scientists and non-data scientists. This is in line with prior work by Jagadish and colleagues which proposed that database query interfaces should strike a fine balance between simplicity and functionality, such that they are simple enough to be learnable for the novice user while still having sufficient functionality to be useful for more technical users [14].

Moreover, this project found that the database query interface facilitated data exploration by reducing disruptions to the user's train of thought. Previous work that aimed to support users at data exploration also worked in a similar manner; for example, progressive visual analytics (PVA) also aimed to increase exploratory speed and enable speed-of-thought analyses by reducing memory and attentional constraints on the user [21, 28, 30]. Additionally, similar to how graphical user interface (GUI) tools supported non-data scientists at data analysis in previous studies [9, 17, 32], this project found that the database query interface empowered non-data scientists and other non-technical users with the ability to independently explore the data and generate their own insights.

Furthermore, some of the challenges faced by users when they were using the database query interface were similar to pain points that were identified in the database query interface literature. For instance, Jagadish et al. also identified that a misaligned mental model in relation to the user flow (Limitation 2) can result in dissatisfaction when users encounter unexpected errors or results, and that poor visibility of system status and feedback (Limitation 3) can frustrate users who were unable to accurately predict the outcome of their queries [14]. Additionally, certain challenges faced by users in this project were in line with some of the barriers faced by non-data scientists during data analysis. In particular, Kwon and colleagues also identified that non-data scientists often needed to be guided toward the appropriate mental models (Limitation 2) and required support for interpreting complex or confusing terms (Limitation 5) [18]. Similarly, Grammel et al. highlighted that data analysis tools need to be able to assist users in error recovery (Limitation 3) and provide clear explanations to users (Limitation 5) in order to successfully support non-data scientists at data analysis [12].

Importantly, this project extends previous work by being the first of its kind to conduct an in-depth investigation into how users interact with a database query interface in a real-world setting. As such, there are several important implications from the results of this project. Firstly, the findings further our understanding of the profiles, goals and pain points of database query interface users in a company (i.e. see Figures 2 and 3 for personas), which provide insights into how database query interfaces can be designed to effectively cater to various user groups. Secondly, the findings highlight the importance of instilling confidence in database query interface users, emphasising the need for designers to come up with solutions to increase user trust in the database query interface (e.g. by incorporating an onboarding process for new users). Together, these findings provide designers and developers with a better understanding of the user requirements of database query interfaces, which can support them in implementing database query interfaces that are learnable and usable.

Designing a User-Centred Database Query Interface

The second aim of this project was to design a database query interface that supports users who engage in regular querying, exploration and analysis of data from large databases. Hence, this project followed an iterative and user-centred design process to create a database query interface that was effective at supporting users in querying customer transaction data from the fast-moving consumer goods (FMCG) industry. The new design incorporated the following seven ideas: (1) a clean visual design, (2) a clear and intuitive user flow, (3) an output preview feature, (4) categorisation of attributes and metrics, (5) tooltips, (6) features for reducing friction and (7) a workspace feature. When the new design was evaluated in usability tests, users were generally pleased with the overall design and the specific ideas, providing mostly positive feedback and some minor suggestions for improvement.

Interestingly, many of the design ideas for the new database query interface were similar to design features that were proposed in previous work. For instance, the clean visual design (Idea 1) and output preview feature (Idea 3) in the new design provided users with visibility of the system status and gave users immediate feedback. Similarly, in the design of GestureQuery, Jiang and colleagues also provided users with instantaneous feedback and followed the principle of "what you see is what you get" (WYSIWYG) [16]. Moreover, the VISAGE database query interface proposed by Zhang et al. had a Query Manager feature that allowed users to re-use saved queries, which was similar to the saved/previous queries feature (Idea 6) that was proposed in this project [34]. VISAGE also had a Query Explorer feature that allowed users to get quick descriptive statistics and generate simple graphs, which was similar to the workspace feature (Idea 7) of the new design. Overall, the findings from this project confirm the effectiveness of these design ideas for enhancing the learnability and usability of database query interfaces.

Crucially, this project extends the database query interfaces literature by being the first to design a database query interface that is meant for frequent use and for querying large databases. In particular, the seven design ideas that were proposed in this project can be translated to a set of design recommendations that can assist designers and developers in creating database query interfaces that are intuitive and user-friendly for both data scientists and non-data scientists. Importantly, increasing the learnability and usability of database query interfaces has important applications for companies who depend on database query interfaces. In general, more non-data scientists and other non-technical users will be able to interact with databases without requiring data scientists, analysts or database administrators to access the data for them, which can empower them with the ability to generate their own insights from the data and reduce the costs required from the company to maintain a technical support team. More specifically, consultancies and agencies such as Dunnhumby who work with external clients will be able to commercialise their database query interfaces, since these tools are now directly accessible to users from client companies who may not have a data science background.

Limitations

There are several limitations present in this project. The first category of limitations was related to the methodology of the user research that was conducted. Firstly, some of the semi-structured interviews and usability tests were not audio- or video-recorded due to privacy reasons. Hence, the researcher was required to take down notes and record quotes from the participants while also conducting the session, which may have resulted in minor disruptions to the flow of the sessions and led to the researcher missing out interesting insights from participants. Secondly, all the participants that were recruited for the interviews and usability tests had some prior experience with the existing database query interface. Even though "potential" users of the database query interface were recruited for the study, these users had some previous encounters with the interface through watching their colleagues use the tool or through demonstration sessions conducted by the company. As a result, this project was unable to investigate how users who were completely new to the database query interface felt about its learnability and usability.

A second category of limitations was related to how the new design of the database query interface was evaluated. Firstly, due to time and resource constraints, the interactive prototype was only of medium fidelity and had limited functionality. In particular, users had restricted access to specific screens and could only navigate to certain sections on each screen. Due to these constraints, the user task during the evaluation had to be specified in detail, and users could perform only the task through a particular user flow. As such, while the prototype was used to evaluate the usability of the new design for building specific queries, it could not be used to evaluate how effective the new design was at communicating the functionality of the database query interface or at assisting users in recovering from errors. Secondly, since the functionality of the prototype was very different from the functionality of the existing database query interface at Dunnhumby, no fair quantitative comparisons (i.e. A/B tests) could be made between the two designs. Thus, the evaluation was only qualitative and did not incorporate any quantitative measures (e.g. number of clicks, completion time). Thirdly, social desirability bias might have been present during the interview since the researcher was also the person who created the new design for the database query interface. Even though participants were reminded to give their honest feedback during the sessions, it was possible that participants had the tendency to provide more positive feedback than negative feedback.

A final limitation of the project was that it was focused on a specific database query interface at a data science company. The database query interface at Dunnhumby was purposefully selected for this project because it was deemed necessary to focus on a particular case study in order to get detailed insights on how employees in a company interact with a database query interface that was used regularly for querying from a large database. Nevertheless, this means that there may be limited generalisability of the findings to other database query interfaces that are used in different contexts and for different types of data, and to other companies where employees might have different roles, workflows and user requirements.

Future Work

A first suggestion for future research is to further examine how different database query interfaces are used in other organisations. In particular, future studies could investigate the practices and workflows of employees from companies of different sizes and industries to observe whether the ways they interact with database query interfaces are similar to what was found in this project. Similarly, future studies could look into different database query interfaces to identify if the limitations found in this project are generalisable to other database query interfaces as well.

A second suggestion for future work is to conduct a more thorough evaluation of the proposed design ideas. In this project, the evaluation demonstrated that the new design was effective at supporting users in building a query when they already had a specific question in mind. However, due to constraints in the functionality of the prototype, this project was unable to evaluate whether the design features were useful at helping users translate abstract research questions into the correct queries, or at assisting users in extracting insights from the query results. Hence, further rounds of usability testing should be conducted using a higher fidelity prototype or Wizard-of-Oz methods to determine if the design ideas were effective at supporting users in data exploration. Additionally, future work should implement the proposed features into a working database query interface to assess whether the design ideas are technically feasible, especially for the output preview feature (Idea 3) and workspace feature (Idea 7).

7. CONCLUSION

Firstly, the empirical contribution of this project was the in-depth investigation into how a database query interface is used in a real-world setting; in particular, the project focused on a particular case study (i.e. a database query interface at Dunnhumby) and examined how employees interacted with this database query interface and the challenges they faced. In particular, this project revealed the use cases of database query interfaces, their usefulness for facilitating data exploration, and the factors that affected user adoption. Additionally, this project identified several design limitations of database query interfaces that make them challenging for users to learn and use. These findings further our understanding of the profiles, motivations and barriers of database query interface users, providing interaction designers with a better understanding of the user requirements of database query interfaces.

Secondly, the design contribution of this project was the design and evaluation of a novel database query interface that supports users who engage in frequent querying, exploration and analysis of data from large databases. By following a user-centred design process, this project came up with a database query interface that was effective at supporting users in interacting with customer transaction data from the fast-moving consumer goods (FMCG) industry. The design artefacts and design ideas generated in this project will assist designers and developers in creating database query interfaces that are learnable and usable for both data scientists and non-data scientists.

ACKNOWLEDGMENTS

I would like to thank my academic supervisor Enrico Costanza for his valuable advice and feedback during the project. I would also like to thank my industry supervisors from Dunnhumby – Jon Geraghty and Rosie Prior – for guiding me throughout the project and ensuring that the project ran smoothly. Additionally, I would like to express my gratitude to the employees at Dunnhumby who volunteered their time to participate in this project. Lastly, I am grateful to my partner, family and friends for their continued support during my academic journey.

REFERENCES

1. Azza Abouzied, Joseph Hellerstein, and Avi Silberschatz. 2012. DataPlay: interactive tweaking and example-driven correction of graphical database queries. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). Association for Computing Machinery, New York, NY, USA, 207–218. DOI:https://doi.org/10.1145/2380116.2380144
2. Marco Angelini, Giuseppe Santucci, Heidrun Schumann, and Hans-Jorg Schulz. 2018. A Review and Characterization of Progressive Visual Analytics. Informatics 5, 3 (June 2018), 31. DOI:https://doi.org/10.3390/informatics5030031
3. Boost Labs. 2021. Many Eyes: IBM's Free Online Data Visualization Tool. Retrieved May 24, 2021 from https://boostlabs.com/blog/ibms-many-eyes-online-data-visualization-tool/
4. Virginia Braun & Victoria Clarke. 2006. Using thematic analysis in psychology. Qualitative Research in Psychology 3, 2 (July 2008), 77-101. DOI:10.1191/1478088706qp063oa.
5. Chartio. 2021. Visual SQL. Retrieved Aug 6, 2021 from https://chartio.com/docs/visual-sql/
6. Elena Demidova, Xuan Zhou, and Wolfgang Nejdl. 2012. FreeQ: an interactive query interface for freebase. In Proceedings of the 21st International Conference on World Wide Web (WWW '12 Companion). Association for Computing Machinery, New York, NY, USA, 325–328. DOI:https://doi.org/10.1145/2187980.2188040
7. Devart. 2021. Query Builder for SQL Server. Retrieved Aug 6, 2021 from https://www.devart.com/dbforge/sql/querybuilder
8. Vasant Dhar. 2013. Data science and prediction. Commun. ACM 56, 12 (December 2013), 64–73. DOI:https://doi.org/10.1145/2500499
9. Micheline Elias and Anastasia Bezerianos. 2011. Exploration Views: Understanding Dashboard Creation and Customization for Visualization Novices. In Human-Computer Interaction – INTERACT 2011 (INTERACT 2011), 274-291. Springer, Berlin, Heidelberg. DOI:https://doi.org/10.1007/978-3-642-23768-3_23
10. Jean-Daniel Fekete and Romain Primet. 2016. Progressive analytics: A computation paradigm for exploratory data analysis. ArXiv Preprint (July 2016), 1–10.
11. Lisa Gitelman. 2006. Raw Data is an Oxymoron. MIT Press, Cambridge, MA.
12. Lars Grammel, Melanie Tory, and Margaret-Anne Storey. 2010. How Information Visualization Novices Construct Visualizations. In IEEE Transactions on Visualization and Computer Graphics 16, 6 (November 2010), 943-952. DOI: https://doi.org /10.1109/TVCG.2010.164.
13. Jeffrey Heer, Frank van Ham, Sheelagh Carpendale, Chris Weaver, and Petra Isenberg. 2008. Creation and collaboration: Engaging new audiences for information visualization. In Information Visualization, 92-133. Springer, Berlin, Heidelberg. DOI:https://doi.org/10.1007/978-3-540-70956-5_5
14. H. V. Jagadish, Adriane Chapman, Aaron Elkiss, Magesh Jayapandian, Yunyao Li, Arnab Nandi, and Cong Yu. 2007. Making database systems usable. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD '07). Association for Computing Machinery, New York, NY, USA, 13–24. DOI:https://doi.org/10.1145/1247480.1247483
15. Magesh Jayapandian and H. V. Jagadish. 2008. Automated creation of a forms-based database query interface. Proc. VLDB Endow. 1, 1 (August 2008), 695–709. DOI:https://doi.org/10.14778/1453856.1453932
16. Lilong Jiang, Michael Mandel, and Arnab Nandi. 2013. GestureQuery: a multitouch database query interface.

Proc. VLDB Endow. 6, 12 (August 2013), 1342–1345. DOI:https://doi.org/10.14778/2536274.2536311

17. Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. 2017. TouchPivot: Blending WIMP & Post-WIMP Interfaces for Data Exploration on Tablet Devices. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). Association for Computing Machinery, New York, NY, USA, 2660–2671. DOI:https://doi.org/10.1145/3025453.3025752
18. Bum C. Kwon, Brian Fisher, and Ji Soo Yi. 2011. Visual analytic roadblocks for novice investigators. In 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), 3-11. DOI:https://doi.org/10.1109/VAST.2011.6102435.
19. Alexandros Labrinidis and H. V. Jagadish. 2012. Challenges and opportunities with big data. Proc. VLDB Endow. 5, 12 (August 2012), 2032–2033. DOI:https://doi.org/10.14778/2367502.2367572
20. Sukwon Lee, Sung-Hee Kim, Ya-Hsin Hung, Heidi Lam, Youn-ah Kang, and Ji Soo Yi. 2016. How do people make sense of unfamiliar visualizations? A grounded model of novice's information visualization sensemaking. IEEE transactions on visualization and computer graphics 22, 1 (January 2016), 499-508. DOI: https://doi.org/10.1109/TVCG.2015.2467195
21. Luana Micallef, Hans-Jorg Schulz, Marco Angelini, Michael Aupetit, Remco Chang, Jorn Kohlhammer, Adam Perer, and Giuseppe Santucci. 2019. The Human User in Progressive Visual Analytics. In EuroVis 2019 - Short Papers, 19-23. The Eurographics Association. DOI:https://doi.org/ 10.2312/evs.20191164
22. Microsoft. 2021. Data Visualisation: Microsoft Power BI. Retrieved May 24, 2021 from https://powerbi.microsoft.com/
23. Kristi Morton, Magdalena Balazinska, Dan Grossman, and Jock Mackinlay. 2014. Support the data enthusiast: challenges for next-generation data-analysis systems. Proc. VLDB Endow. 7, 6 (February 2014), 453–456. DOI:https://doi.org/10.14778/2732279.2732282
24. Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, Paper 126, 1–15. DOI:https://doi.org/10.1145/3290605.3300356
25. Samir Passi and Steven J. Jackson. 2018. Trust in Data Science: Collaboration, Translation, and Accountability in Corporate Data Science Projects. Proc. ACM Hum.-Comput. Interact. 2, CSCW, Article 136 (November 2018), 1–28. DOI:https://doi.org/10.1145/3274405
26. Tate A. Rosemary, Natalia Beloff, Balques Al-Radwan, Joss Wickson, Shivani Puri, Timothy Williams, Tjeerd Van Staa, and Adrian Bleach. 2014. Exploiting the potential of large databases of electronic health records for research using rapid search algorithms and an intuitive query interface. Journal of the American Medical Informatics Association 21, 2 (March 2014), 292-298. DOI:https://doi.org/10.1136/amiajnl-2013-001847
27. Saeed Shahrivari. 2014. Beyond Batch Processing: Towards Real-Time and Streaming Big Data. Computers 3, 4 (October 2014), 117-129. DOI: https://doi.org/10.3390/computers3040117
28. Charles D. Stolper, Adam Perer, and David Gotz. 2014. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. In IEEE Transactions on Visualization and Computer Graphics 20, 12 (December 2014), 1653-1662. DOI:https://doi.org/10.1109/TVCG.2014.2346574.
29. Tableau. 2021. Business intelligence and analytics software – Tableau Software. Retrieved May 24, 2021 from https://www.tableau.com/
30. Cagatay Turkay, Erdem Kaya, Selim Balcisoy, and Helwig Hauser. 2017. Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. In IEEE Transactions on Visualization and Computer Graphics 23, 1 (January 2017), 131-140. DOI: https://doi.org/10.1109/TVCG.2016.2598470.
31. Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. Proc. ACM Hum.-Comput. Interact. 3, CSCW, Article 211 (November 2019), 24 pages. DOI:https://doi.org/10.1145/3359313
32. Mehmet A. Yalcin, Niklas Elmqvist, and Benjamin B. Bederson. 2018. Keshif: Rapid and Expressive Tabular Data Exploration for Novices. In IEEE Transactions on Visualization and Computer Graphics 24, 8 (August 2018), 2339-2352. DOI:https://doi.org/10.1109/TVCG.2017.2723393.
33. Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. Proc. ACM Hum.-Comput. Interact. 4, CSCW1, Article 022 (May 2020), 23 pages. DOI:https://doi.org/10.1145/3392826
34. Guo-Qiang Zhang, Trish Siegler, Paul Saxman, Neil Sandberg, Remo Mueller, Nathan Johnson, Dale Hunscher, and Sivaram Arabandi. 2010. VISAGE: A query interface for clinical research. Summit on Translational Bioinformatics (March 2010), 76-80.