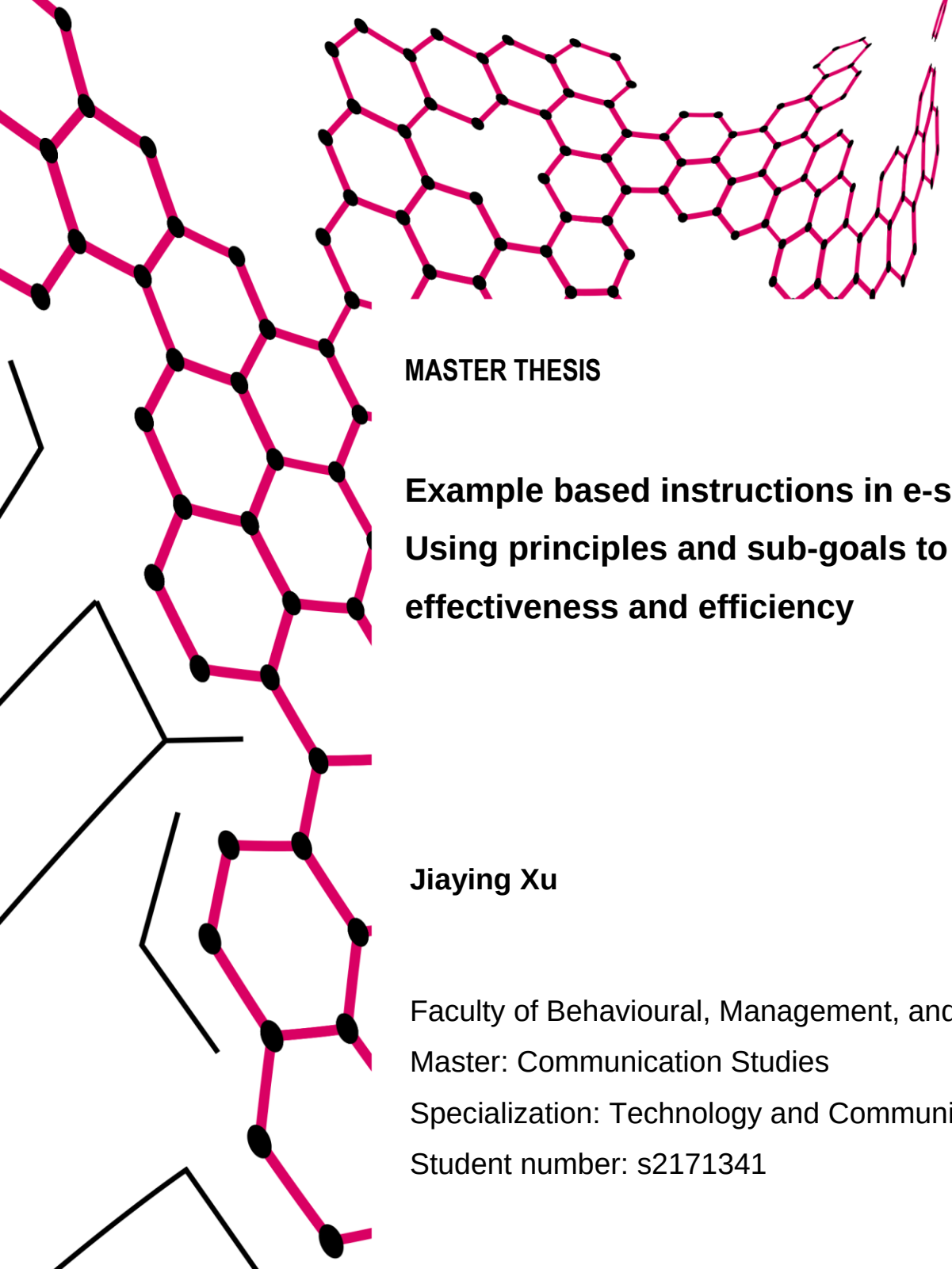

MASTER THESIS

Example based instructions in e-science: Using principles and sub-goals to improve effectiveness and efficiency

Jiaying Xu

Faculty of Behavioural, Management, and Social Sciences
Master: Communication Studies
Specialization: Technology and Communication
Student number: s2171341

First Supervisor: **Dr. J. Karreman**
Second Supervisor: **R. S. Jacobs PhD**

UNIVERSITY OF TWENTE.

Acknowledgement

I would like to thank my supervisors, Dr. Joyce Karreman and Dr. Ruud Jacobs, for their guidance, support and patience. I would also like to show my gratitude to all 102 participants. Without their help, I could not have finished the experiment within limited time. Finally, I want to thank my family and my boyfriend Yuming, who encouraged me to do this study.

Abstract

Purpose: Examples are widely used in procedural instructions, especially in complex domains such as e-science in which scientific knowledge and technical information are intertwined. Example-based instructions are effective because users tend to follow the instantiated and specific procedures. Prior studies have pointed out that adding sub-goals to examples can benefit users' task performance and efficiency because it makes sub-tasks easy to understand and memorize. Principles also have positive influences on users' learning and transfer of knowledge as they enable users to generalize their learning outcomes. This study is designed to determine whether the general principles and specific sub-goals are effective in examples of e-science as proven by previous research in the examples of other contexts.

Method: A 2*2 between-subjects experiment was conducted with presence of sub-goals and principles as independent variables, and the effect on task efficiency and effectiveness as dependent variables. Novice users of sky survey databases were selected as the target group, and 93 participants were assigned to four conditions with different versions of instructions. Each participant was asked to complete ten learning tasks and five transfer tasks. Their task performance and task efficiency were measured.

Results: Users showed better performance in learning tasks and higher efficiency in transfer tasks when assisted by examples with sub-goals. No significant difference was found in transfer performance and learning efficiency between groups with and without sub-goals, nor between groups with and without principles.

Conclusion: The effectiveness of sub-goal information was proven in the context of e-science databases. There is no significant effect regarding general principles when added to examples. Therefore, in documentation practices, descriptions of sub-tasks are suggested for use in examples in order to reduce complexity. More research is needed to study principles and sub-goals by improved approach and to focus on cognitive and emotional effects.

Keywords: e-science, user instructions, examples, principles, sub-goal labels

Table of Contents

1. Introduction

The internet became an important tool for education, business and science in the 1990s. Then, after several years of development, the scientific world started to explore the potential of linking computers and integrating computing resources so as to access data and experimental equipment from remote sites. This created the concept of e-science, which is considered a special infrastructure of various forms such as search engines, specialized databases, and data-mining tools. E-science applications enable data-intensive research, projects across distributed data repositories and collaborations between scientific communities and institutes (Hey & Trefethen, 2005). One typical example of e-science is the Large Hadron Collider (LHC) Computing Grid in Geneva, which aims to find experimental evidence of the Higgs particle. At LHC, large-scale experiments are conducted and a huge amount of data generated every year, so analysis and simulation can only be done through collaborations between scientists from different countries. Nowadays a number of similar projects have been implemented in various disciplines such as geoscience, environmental science and astronomy.

As have been widely deployed, e-science infrastructures also created needs for management and user support. Users of e-science infrastructure may include experts, novice scientists and students with various technical skill levels. Among these skills, writing data queries according to structured query language (SQL) is one of the most common ways to retrieve scientific data and manipulate databases. Hence, providing web-based technical content to support complex SQL usage becomes increasingly important. From users' perspectives, efficiently composing the right queries is an indispensable procedure in doing science. However, from e-science developers' perspectives, SQL query logs, in other words the log files containing what users submit, have been employed as a method to investigate user behavior and improve user experience (Makiyama, Raddick & Santos, 2015). If users have access to sufficient and effective user instructions, SQL formatting errors can be avoided so that query-log analysis will become easier.

When utilizing SQL instructions, e-science users' learning process can be intertwined with

connecting disciplinary concepts with database parameters and specified functions that represent them, which results in the wide use of query samples. These samples are mostly structured as worked examples, so users can copy the given examples and test or modify them to fulfill their own goals. This self-learning behavior can be explained by theoretical works of example-based learning and the four-stage skill acquisition model (Anderson, Fincham & Douglass, 1997). Though some e-science applications have developed web search forms for novices, many of their advanced functions can only be used when searching by SQL query code. Therefore, SQL query samples often serve as instructive agents in learning. Similarly, in the software industry, working examples of code or data to use with the product is one of the major genres of software documentation used by over 50% of users on a weekly or monthly basis (Earle, Rosso & Alexander, 2015).

Given this need to learn complex queries through examples, their design and structure play a crucial role in learning performance and learners' experience (Atkinson, Derry, Renkl & Wortham 2000; Chen, Mitrovic & Matthews, 2019). Novice users of e-science databases especially rely on query examples both offered in documentations and embedded in query interfaces. Through these examples they become familiar with the tables and fields defined in each specific database, and prepare to perform complex tasks in which multiple databases may be involved. The effectiveness and efficiency of example-based learning contribute to the overall image and quality of e-science SQL documentation.

Given that most research have studied example-based learning in classroom settings, this method also has its effectiveness in informal learning such as training and skill acquisition (Atkinson, Derry, Renkl & Wortham, 2000). To further test their effect, examples of various forms have been explored in past three decades, such as worked examples, erroneous examples, and sub-goal example. The usefulness of examples assisted by information in different levels of specificity were compared by Catrambone (1995) and Morrison, Margulieux and Guzdial (2015), who attempted to study intra-example features such as sub-goals and principles.

This study firstly summarizes prior studies on user instructions in e-science and effects of different

examples in the theoretical framework. Then, a 2*2, between-subject experiment is described to investigate the effectiveness and learning efficiency of examples structured differently by using principles and sub-goals. Through quantitative analysis, it attempts to discover the best practice of designing examples for novice users of e-science databases. This may give insight into the distinct challenges faced by e-science user support.

Research question:

How do SQL query examples, as specific instructions in e-science databases, combined with general principles and sub-goal labels to improve users' performance and efficiency?

2. Theoretical framework

This chapter consists of two parts. The first part focuses on user support in e-science, especially for SQL usage. Then effects of example-oriented instructions and two types of information, principles and sub-goals, are explained in the second part.

2.1. Theoretical basis of supporting e-science users in SQL data query

2.1.1. Supporting users of e-science applications

E-science infrastructures have covered many data-intensive areas of science such as astronomy, geoscience and physics. Supporting users, namely researchers, to complete their scientific tasks and activities is a vital part of facilitating research based on e-science infrastructure. In the projects that have been studied in recent years, user support of e-science is offered mainly in two forms: self-help (e.g. websites, online tutorials and wikis), and traditional help desks (e.g. mailing lists, human support agents) (Chunpir, Ludwig & Curri, 2014). Besides, user communities, face-to-face training and workshops organized by research institutes can additionally be listed as user support procedures.

The differences between user support in e-science and user support in industry have been recognized and pointed out by Chunpir, Ludwig and Curri (2014) and Swarts (2019): (1) in the academic setting,

user needs often relate to both scientific and technical problems; (2) these problems are to be solved by the staffs of geographically distributed institutions; (3) representation of disciplinary knowledge can be a challenge to user support designers. Most prior studies focused on the interaction between support staffs and users as well as organizational strategies (Chunpir, Ludwig & Curri, 2014; Chunpir & Ludwig, 2014). The user experience and user interface design in e-science came as a new focus (Chunpir, 2018).

However, how the documentation practices recommended by literature and IT industries can be applied to improve the quality of web-based help content for e-science tools is a question to be answered and the gap is to be filled by empirical studies. In e-science applications, the acquisition of technical skills and recall of scientific knowledge both seem important in user cognition. Whether or not current approaches, for example, minimal user instructions, design features supporting usability or selecting information types on a user-centered basis remain effective has not yet been tested in this context.

2.1.2. Facilitating SQL usage in e-science databases

Structured query language is a programming language used to manage data in relational databases in which data are structured and relations are built between entities or variables. It has become an effective method to work with large datasets and contributed to several successful e-science projects including Sloan Digital Sky Survey (SDSS) and Gaia in the field of astronomy, ArcGIS in the field of geography and SQLShare as a general data platform. An increasing number of queries and a growing user community have been observed according to SQL logs analysis of SDSS, and they indicate the effective usage of SQL query and changing user manner. Although users may start with web-form-based searches, these scientists switch gradually to composing query code and using predefined functions rather than filling in search forms when they gain a proficiency in SQL (Li & Thakar, 2008).

User interactions with scientific data through SQL can be divided into two phases: a query

composition phase, in which users compose and submit a query; and a query answer analysis phase, in which users view and analyze the results produced by database system (Zolaktaf, 2017). Composing useful queries and selecting relevant data from database systems is still a challenge faced by scientists at the first stage. In the worst case, researchers spend almost nine times as much time on technical problems such as managing data as they spend on scientific problems (Howe, Cole, Khoussainova & Battle, 2011). Although scientists and other e-science users are less likely to acquire significant SQL expertise through formal curriculum, offering a rich set of query examples empowers them to use SQL for data retrieval (Howe, Cole, Khoussainova & Battle, 2011). The successful practices of SDSS and many other scientific databases have demonstrated this strategy by including examples in their documentation.

Furthermore, providing technical support for scientific SQL users by various approaches is necessary because inexperienced usage and mistaken queries can be a danger to a scientific database. If users fail to plan their queries carefully, it may cause a large amount of network traffic and interfere with other users' activities. In the case of Skyserver, the primary catalog data portal of SDSS, a significant number of queries are prematurely terminated by the server because of wrong conditions or parameters chosen by inexperienced users (Ivanova, Nes, Goncalves & Kersten, 2007). Limits on time of running a search and the number of records can be retrieved by one query should also be noted by users. The server may not return anything if the amount of data they try to search is too large even if they submit the right code.

In current practices of e-science databases, types of SQL documentation include tutorials, how-to guides, reference manuals and searching advice,e while all databases provide sample queries, namely examples. These examples generally contain goal descriptions, tables used and a query code block. Special information types used in particular databases include utilization information, sub-goal description, system information, syntax explanations and query results (see Appendix 1). This raises a practical question: which types of information in written instructions are more effective in aiding scientific users' performance of using SQL?

2.2. Theoretical basis of example-oriented instructions

2.2.1. Types of information in user instructions

As people often encounter procedural tasks in their daily life, they may have to use new tools, technologies and systems that they are not familiar with. Being instructed by written materials is a common method in various situations. These instructions contain different types of information, such as procedural action steps, general rules and specific examples. The definition given to procedural information, which is information that describes user actions, conditions for actions and results from actions, by Ummelen (1997) has been widely accepted and used in present studies. However, the classification of other information types varies according to different researchers. Ummelen (1997) only made the distinction between procedural information and all types of explanatory information other than procedural information, namely declarative information. Declarative information was further categorized by Karreman (2004) as system information about the internal working mechanism of the product and utilization information about the conditions in which a particular function can be used. Another way of classification was proposed by Eiriksdottir and Catrambone (2011) who excluded utilization information about the current state of a system or environment where the task is to be carried out from instructions. Besides the most important type, procedural instructions, they distinguished two major types of instructions: principles which are general rules governing the task domain, and examples demonstrating how the task is actually performed.

Although researchers have different ways of categorizing instructions, all these information types are respectively studied and widely used in real instruction practices. Besides other explanatory information added to procedural information, the specificity of procedural instructions also has an effect. A tradeoff between general and specific instructions was recognized by Catrambone (1990), as he found detailed instructions restricted to certain cases help users apply the learning gains immediately but could be problematic in further transfer. On the contrary, general instructions that are able to support a wider range of tasks are more helpful for generalizing the procedure learned but users reported more difficulty in initial use. Catrambone (1995) examined two strategies for

improving general instructions by adding principles or examples and the results showed that initial performance was improved in both conditions so that the drawback of general instructions is compensated. In addition, although reading longer text requires more time when principles or examples are added, users become more efficient at comprehending the action steps, so the overall time spent is balanced. As for specific instructions, adding general instructions or principles has an obvious effect on generalization when users get to novel situations even if specific instructions have been rewritten to support this goal (Catrambone, 1990). Catrambone also pointed out that it is worth investigating the effect of combining the features of general and specific instructions, but using examples as specific instructions has not yet been explored from this perspective.

H1. Users of e-science databases will spend more time reading learning materials when using query examples with sub-goal or principle labels than users using examples without these labels.

2.2.2. Use and effect of examples

Examples illustrate how abstract and complex instructions can be instantiated and allow users to gain a better understanding of general instructions or principles (Eiriksdottir & Catrambone, 2011). It is still considered a special information type because whether examples are classified as procedural information or declarative information largely depend on the particular content. In previous studies, examples generally referred to descriptions of how the task is performed under certain conditions. Thus, they are regarded as procedural information by Karreman (2004) while Eiriksdottir and Catrambone (2011) later pointed out that the difference between procedural instructions and examples lies in whether the task is described or demonstrated. In this research, considering the real SQL examples used in e-science context, examples are classified as a type of specific procedural instruction, as blocks or lines of SQL statement are essential components in any SQL manual.

Prior studies have proven the usefulness and effectiveness of using examples in instructions. The example effect was found by LeFevre and Dixon (1986): examples were more compelling than written instructions, and there was a strong tendency to follow the example. This effect was proven

to be robust and pervasive in various conditions where examples and written instructions were presented in different orders or with descriptions of incorrect usage. Students and novice scientists presented with worked examples showed better problem-solving performance and strategies compared to using a traditional, practice-based method (Sweller & Cooper, 1985). This may be because the similarity of content and surface features between examples and novel tasks contributes to extracting action steps and generating good initial task performance. Trying to relate known examples to the problem is a typical analogy process in the initial stage of skills acquisition, as in the theoretical model proposed by Anderson, Fincham, and Douglass (1997).

Using examples in web-based instructions also has its potentials and helps researchers gain insights into user behavior. An example-based approach for online documentation design, especially for non-linear tasks in solving complex problems, was proposed by Tomasi and Mehlenbacher (1999). Examples were designed for the learning process of specific concepts needed for using a software, and users who were experienced software developers with relevant skills were aided by these examples to become productive in a brief time. As in the software setting examples are presented in the form of complete code files, modes of user interaction with examples included (1) modifying the provided examples to meet specific needs, (2) directly copying features from examples into their own projects and (3) operating the original and unmodified examples. Theses ways of interaction indicated the usefulness and effectiveness of example-based technical support and revealed the potential for how structured examples can be integrated in user documentation and benefit information products such as custom data entry and office automation tools.

2.3. Use and effect of principles

Principles, or in other words system information, are general rules governing the entire system that users work with. It is commonly assumed that principles contribute to the understanding of the system and the construction of a mental representation. However, it has been controversial whether principles should be included in instructions. The minimalism approach highlights the task-oriented nature of actions and largely ignores this type of information while some research has shown users'

interest and needs in reading conceptual knowledge (Karreman, Ummelen & Steehouder, 2005).

Experiments did show that users read both system information and utilization information while no significant effect on cognitive load, confidence, knowledge, or even negative influence was shown in the aspect of appreciation toward the device and the instructions (Ummelen,1997; Karreman, 2004). A number of studies also aimed to investigate the influences of system information on learning and transfer, but the results have been inconclusive so far. Smith and Goodman (1984) found that learners were able to perform tasks more efficiently when fundamental principles about steps were provided. Catrambone (1995) found that adding principles could aid transfer performance, because users were able to reason more effectively about new tasks. However, it was not clear whether or not principles were able to eliminate the difficulty of generalizing examples. While these studies showed positive outcomes, others showed no obvious effect (Karreman & Steehouder, 2004; Kieras & Bovair, 1984; Reder, Charney & Morgan, 1986). This might result from a diversity of instructions provided together with principles. In general, users are able to benefit from principles and gain better performance in learning and transfer, but how much they rely on and make use of principles is influenced by how detailed other instructions are (Duff & Barnard, 1990; Eiriksdottir, 2011). The fewer detailed instructions provided, the more they follow the principles. It was suggested by Duff and Barnard (1990) that people are forced to learn from principles and practice with them when no specific instructions are accessible so as to develop mental representation and gain better performance in transfer tasks. From this perspective, the lack of obvious advantage found in prior studies may result from presenting highly detailed procedural information along with principles.

H2a. Users of e-science databases will be more efficient in transfer of learning when using query examples with principles than users using examples without this information.

H3. Users of e-science databases will have better transfer performance when using SQL query examples with principle labels compared to users using examples without this type of information.

2.4. Use and effect of sub-goals

As examples have been recognized as effective instructional devices, how examples are structured becomes important to encourage users to employ better strategies in enhancing transfer and learning performance. The design features, relationships between multiple examples and practices, and ways of self-explaining are suggested by Atkinson, Derry, Renkl and Wortham (2000) as factors determining the effectiveness of examples. Emphasizing sub-goals is one of the mainly studied structuring approaches in enhancing example-based learning. Sub-goal labels are given to a set of actions or steps in order to highlight shared features within the task and enhance transfer of learning. Sub-goals eliminate the difficulty of memorizing the whole procedure with little meaning explained and completing new tasks with changed action steps. With better understanding and memory of examples that have been learned, learners are capable of applying sub-steps to performing similar learning tasks. The sub-goal model assumes that labeling sub-goals enables learners to engage in self-explanation and promote their understanding of task structure and procedures (Catrambone & Holyoak 1990; Catrambone, 1998). Empirical studies have shown that adding structural cues, such as affixing a label to sub-goals within a solution or visually isolating them, improved transfer of learning, as more effective self-explanations were observed through think-aloud approach. In addition, Morrison, Margulieux and Guzdial (2015) proved that adding sub-goal labels results in more efficient problem solving in learning tasks.

H2b. Users of e-science databases will be more efficient in transfer of learning when using query examples with sub-goals than users using examples without this information.

H4. Users of e-science databases will have better initial learning performance when using query examples with sub-goal labels compared to users using examples without this type of information.

2.5. Hypotheses

Since research about structuring SQL query examples in e-science databases is limited, this study explores the structuring strategy of presenting examples with sub-goal and principle labels to support better learning performance and a more friendly learning environment. Here, the research

question is how SQL query examples, as specific instructions in e-science databases, work with general principles and sub-goal labels to improve users' performance and efficiency? Five hypotheses based on prior research are listed as below. Besides, interaction effects of two independent variables, principles and sub-goals, may exist in users' learning performance and transfer performance (see ***H5*** and ***H6***). Though these two effects are not expected based on prior studies, examination of the interaction effect is included in this research. It is not only because some possible simultaneous effects of principles and sub-goals have never been found, but also because the absence of interaction effect provides evidence for generalization of a single variable's effect.

Time efficiency

H1. Users of e-science databases will spend more time reading learning materials when using query examples with sub-goal or principle labels than users using examples without these labels.

H2a. Users of e-science databases will be more efficient in transfer of learning when using query examples with principles than users using examples without this information.

H2b. Users of e-science databases will be more efficient in transfer of learning when using query examples with sub-goals than users using examples without this information.

Effectiveness

H3. Users of e-science databases will have better **transfer performance** when using SQL query examples with principle labels compared to users using examples without this type of information.

H4. Users of e-science databases will have better **initial learning performance** when using SQL query examples with sub-goal labels compared to users using examples without this type of information.

Possible interaction effects:

H5. Because of the interaction effect, users of e-science databases will have better **initial learning performance** when using SQL query examples with sub-goal labels and principles compared to users using examples without two types of information.

H6. Because of the interaction effect, users of e-science databases will have better **transfer**

performance when using SQL query examples with sub-goal labels and principles compared to users using examples without two types of information.

3. Methods

3.1. Experimental design

Based on prior studies about the effect of examples, a quantitative method was employed to investigate how principles and sub-goals influence the effectiveness of examples among e-science users. Astronomy and large sky survey database were chosen as the specific context and e-science application in this study. A 2*2 between-subject experiment was conducted with two independent variables (sub-goals and principles) and two subclasses of each variable (absence and presence). Differences between examples in four conditions are listed in Table 3-1.

Table 3-1 *Four conditions in the experiment*

Condition	With principles	With sub-goals
A	No	No
B	Yes	No
C	No	Yes
D	Yes	Yes

3.2. Participants

To test the effect on novice users of e-science database, participants were required to have limited experience or no experience with SQL. 93 participants met this requirement and completed the experiment successfully. They are all astronomy majors in China while 56 of them are males and 37 are females. They were randomly assigned into four groups, with 24 students in group D and 23 in each of other three groups, to perform the task under four different conditions respectively. There was no significant difference between four groups regarding the correctness of 10 answers in knowledge test (see Appendix 3): $F(3,99) = .78$, $p = .508$. Therefore, the levels of prior SQL knowledge among different groups are considered as same. Mean scores of the knowledge test are

shown as below:

Table 3-2 *Four groups' performance in the knowledge test*

Group	Mean score of correctness
A (no principle, no sub-goal)	0.68
B (principles, no sub-goal)	0.54
C (sub-goals, no principle)	0.84
D (principles, sub-goals)	0.96
Total	0.75

There is no significant difference of average age among four groups: F (3,89) =1.410, p=.245. Education level and average age of participants are shown below:

Table 3-3 *Education level and average age of participants*

Group	Bachelor	Master	PhD	Average age
A (no principle, no sub-goal)	13	8	2	22.39
B (principles, no sub-goal)	12	7	4	23.96
C (sub-goals, no principle)	16	5	2	22.87
D (principles, sub-goals)	13	8	3	22.75
Total	54	28	11	22.99

3.3. Materials

3.3.1. Learning materials

The examples in this study were selected from Sloan digital sky survey (SDSS). In large sky survey databases operated by different institutes, SQL query examples are presented with various types of declarative information such as goal description, use case, explanations of SQL syntax and each command line while different types of information are not categorized or labeled. In this study, four different SQL example books were designed in which selected examples were structured into four conditions by the researcher. A general introduction to SQL statement and tables involved in the material was added. The content covered basic syntax and queries regarding tables that are often used, which was sufficient to instruct novice users to comprehend functions and start composing some code. Besides the original instructions in English, the general introduction and statement of goals in each example was translated into Chinese and offered to participants in order to assist them

in recalling astronomical terms they have learned in Chinese. (See Appendix 2)

To be more detailed, each of the four conditions had different types of information to support users of these query examples. In condition A, queries were not aided by additional information just like most existing documentations. Examples in condition B and C were combined with principle labels or sub-goal labels respectively, and condition D had both two types of label. The comparison of four versions is as in the next page:

Table 3-4 *An example in four conditions*

Condition	**Features of query examples**	**Example**
A (no principle, no sub-goal)	Statement of final goal; statement of tables used; code	- Count the number of spectra of each spectral classification (galaxy, quasar, star). - Table used: SpecObj SELECT class, count(*) FROM SpecObj GROUP BY class
B (principles, no sub-goal)	Statement of final goal; statement of tables used; code; principles labeled as in-line notes	- Count the number of spectra of each spectral classification (galaxy, quasar, star). - Table used: SpecObj SELECT class, count(*) -- The count(*) statement returns the number of all records that meet specific search criteria FROM SpecObj GROUP BY class -- The GROUP BY statement groups results into groups (categories) based on the value of a data column.
C (sub-goals, no principle)	Statement of final goal; statement of tables used; codes; sub-goals labeled as in-line notes	- Count the number of spectra of each spectral classification (galaxy, quasar, star). - Table used: SpecObj SELECT class, count(*) -- to retrieve class and the total number of all records FROM SpecObj GROUP BY class -- to group the number of records based on 'class' column which contains the spectral classification of the object
D (principles, sub-goals)	Statement of final goal; statement of tables used; codes; sub-goals and principles labeled as in-line notes	- Count the number of spectra of each spectral classification (galaxy, quasar, star). - Table used: SpecObj SELECT class, count(*) -- to retrieve class and the total number of records -- The count(*) statement returns the number of all records that meet specific search criteria FROM SpecObj GROUP BY class -- to group the number of records based on 'class' column which contains the spectral classification of the object -- The GROUP BY statement groups results into groups (categories) based on the value of a data column.

3.3.2. Survey

After learning the examples, participants started to do Exercise 1 and Exercise 2 in a survey form to practice what they just learnt (see Appendix 4). Exercise 1 consisted of 10 learning tasks with which participants answered five single choice questions about general principles and performed five labeling tasks by highlighting the given query line. Then Exercise 2 presented transfer tasks, which are new queries selected from SDSS and using the same SQL statements as in the learning materials while the goals were slightly different. Participants were asked to perform five labeling tasks by highlighting query lines that serves the goal stated by the question stem. In both Exercise 1 and Exercise 2, participants could choose 'I don't know' if they did not know which answer was correct. Examples of survey questions are listed as in the next page:

Table 3-5 *Survey questions*

	Question type	**Task type**	**Question examples**
Exercise 1	Single choice question	Learning task	With SQL, which clause would you use to sort the results? A. ORDER B. SORT C. ORDER BY D. SORT BY E. I don't know
	Highlight question	Learning task	In the following query, could you highlight the clause(s) that specify(ies) the criteria for smilar spectra (colors)? SELECT TOP 10 P.ObjID FROM PhotoPrimary AS P JOIN Neighbors AS N ON P.ObjID = N.ObjID JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID WHERE P.ObjID < L. ObjID and abs((P.u-P.g)-(L.u-L.g))<0.05 and abs((P.g-P.r)-(L.g-L.r))<0.05 and abs((P.r-P.i)-(L.r-L.i))<0.05 and abs((P.i-P.z)-(L.i-L.z))<0.05 I don't know
Exercise 2	Highlight question	Transfer task	When using SDSS DR15 Image List Tool, all Image List queries must have a clause to retrieve coordinate pairs of objects. In the following query, could you highlight the clause(s) used for this function? SELECT TOP 100 P1.objID AS name, P1.ra AS ra, P1.dec AS dec, FROM PhotoTag P1, Neighbors N, PhotoTag P2 WHERE P1.objID = N. objID AND P2.objID = N.NeighborObjID AND N.Distance < 0.05 I don't know

3.3.3. Pre-test

Nine students from the target group were asked to pretest the learning materials to help estimate the difficulty of SQL syntax involved and time spent on using learning materials. They did not participate in the main experiment afterwards. Besides, all instructions and four versions of the survey were checked and tested with a real database user, another student who are experienced with this database, to avoid misunderstanding and ambiguous wording. As SDSS is restricted to optical sky survey, Chinese translations of specific astronomical terms were added according to the needs of students from different subdomains of astronomy. This avoided misunderstanding of scientific goals and allowed participants to concentrate on acquiring technical skills.

3.4. Measurements

Two components of performance in learning from four different materials were measured by the survey: effectiveness and efficiency. In evaluating effectiveness, participants' learning outcome and transfer of knowledge were measured by the number of questions they answered correctly in Exercise 1 and Exercise 2. In single choice tasks, giving a wrong answer or choosing 'I don't know' were both regarded as not performing the task correctly. And in highlight tasks, participants were required to choose the exact clauses that are relevant to sub-goals or principles. Choosing wrong clauses or more than the right clause(s) is regarded as giving incorrect answers. Besides, users' efficiency in four conditions were compared based on time spent during learning and filling in the two exercises, which was counted by the timer embedded in the survey platform.

3.5. Procedures

The experiment was computerized and based on online instructions and online survey. It consisted of three sessions: a preparation session during which participants signed the consent form, and test their prior knowledge of SQL, a learning session where examples were presented to participants along with a basic introduction to the database, and a testing session. In the learning session, four groups of participants learnt from examples in the four conditions respectively. Then each of them

performed two groups of tasks in the testing session. An overview of three sessions is given by the figure.

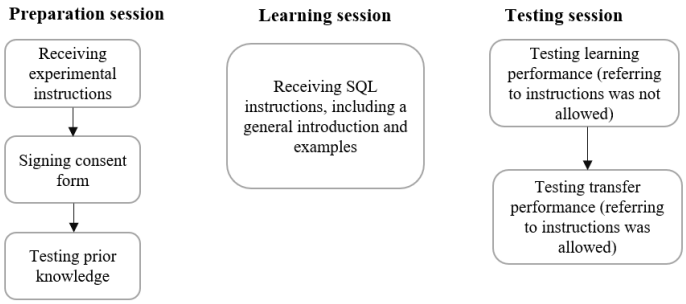

Figure 1. Overview of experimental procedures

4. Results

This section presents results and findings of the experiment. Differences of effectiveness and time efficiency between four versions of instructions are described.

4.1. Differences in learning performance

Among 93 participants, two participants failed to complete more than one learning task so that their data is detected as outliers by descriptive statistics. The mean scores of 91 valid data records are shown in Table 4-1.

Table 4-1 *Difference in learning performance*

Principles	Sub-goals	*M*	*SD*	N
No	No	6.22	2.11	23
	Yes	7.82	1.87	22
	Total	7.00	2.13	45
Yes	No	6.57	2.52	23
	Yes	7.74	1.84	23
	Total	7.15	2.26	46
Total	No	6.39	2.30	46
	Yes	7.78	1.83	45
	Total	7.07	2.19	91

By two-way ANOVA tests, there is no effect of adding principles on learning performance: F (3, 87) = .09, p =.761. The difference of learning performance between groups with and without sub-goal labels is significant: F (3, 87) =9.88, p=.002. Sub-goals have a positive effect with medium effect size ($\hat{\eta}_p^2 = 0.102$). No interaction effect between principles and sub-goals was found: F (3, 87) =.23, p =.630.

4.2. Differences in transfer performance

Transfer performance is measured by five highlight tasks. Also, only highlighting the same clause(s) as mentioned by principles or sub-goals is counted as a correct answer. The mean scores of 93 participants' transfer performance are shown in Table 4-2.

Table 4-2 *Difference in transfer performance*

Principles	Sub-goals	*M*	*SD*	N
No	No	2.13	1.42	23
	Yes	2.35	1.30	23
	Total	2.24	1.35	46
Yes	No	1.87	1.36	23
	Yes	2.08	1.18	24
	Total	1.98	1.26	47
Total	No	2.00	1.38	46
	Yes	2.21	1.23	47
	Total	2.11	1.31	93

Although examples with sub-goals score a little higher than examples without this information, there is no significant effect on transfer performance between them: F (3, 89) = .62, p= .432. It also shows no effect of principles on transfer tasks performance: F (3, 89) = .93, p= .339. There is no interaction effect between principle labels and sub-goal labels in assisting participants to complete transfer tasks: F (3, 89) =.00, p= .995.

4.3. Differences in time efficiency

Time spent in reading materials in the first phase of this experiment was recorded, and 19 of 93 records are excluded from the data set as outliers. 16 outliers of extremely low values, ranging from 5.83 seconds to 88.52 seconds, may result from procedural failure as participants closed the survey

webpage by mistake when reading the example webpage. Besides, other three outliers that are over 6,000 seconds are excluded based on descriptive statistics. The mean scores of 74 valid values are shown in Table 4-3.

Table 4-3 *Difference in learning time*

Principles	Sub-goals	*M*	*SD*	N
No	No	1011.09	843.59	19
	Yes	811.68	632.26	15
	Total	923.11	753.57	34
Yes	No	767.86	811.81	21
	Yes	703.45	694.38	19
	Total	737.26	749.38	40
Total	No	883.39	825.56	40
	Yes	751.20	659.97	34
	Total	822.65	751.94	74

No significant effect on learning efficiency is found between materials with and without principles: $F(3, 70) = .98$, $p = .326$. And no significant effect is found between materials with and without sub-goals: $F(3, 70) = .55$, $p = .460$. Besides, there is no interaction effect between sub-goals and principles: $F(3, 70) = .14$, $p = .705$. Though adding different types of information to examples seems helpful to lessen participants' efforts invested in reading and learning, it does not obviously influence the time duration.

Time spent in performing learning tasks was recorded. Four outliers that are higher than 1,200 seconds are excluded based on descriptive statistics. Mean scores of 89 valid values are shown in Table 4-4.

Table 4-4 *Difference in efficiency of learning tasks*

Principles	Sub-goals	*M*	*SD*	N
No	No	501.43	231.80	22
	Yes	376.63	164.34	22
	Total	439.03	208.36	44
Yes	No	433.23	182.63	22
	Yes	418.08	194.79	23
	Total	425.48	186.94	45
Total	No	467.33	209.09	44
	Yes	397.82	179.72	45
	Total	432.18	196.81	89

The impact of adding principles on learning task efficiency is not significant: F (3, 85) = .11, p= .747. The effect of sub-goals on how much time participants used in learning tasks is not significant: F (3, 85) =2.87, p= .094. Besides, there is no interaction effect found between two types of labels added to the four conditions: F (3, 85) =1.759, p= .188.

Table 4-5 *Difference in efficiency of transfer tasks*

Principles	Sub-goals	*M*	*SD*	N
No	No	528.30	237.18	22
	Yes	433.05	249.21	22
	Total	480.67	245.21	44
Yes	No	551.33	323.25	23
	Yes	372.48	219.42	24
	Total	460.00	286.73	47
Total	No	540.07	281.48	45
	Yes	401.45	233.51	46
	Total	467.00	266.19	91

In time spent in performing transfer tasks, two outliers are excluded by descriptive statistics. Among 91 valid results of transfer task duration, impact of principles is not significant: F (3, 87) = .12, p= .732. Use of sub-goals is effective for improving efficiency in transfer tasks: F (3, 87) =6.30, p=.014. There is no interaction between two independent variables: F (3, 87) =.59, p=.446.

5. Discussion and conclusion

5.1. Main findings

There are two major findings in this study. First, adding sub-goals to examples improved initial learning performance among novice users of e-science databases. Second, on the question of efficiency, using examples aided by sub-goals resulted in shorter time spent in performing transfer tasks. To determine the impact of using different examples on effectiveness and efficiency, seven hypotheses were put forward based on prior studies and possible interaction effects. The results of these hypotheses are summarized in the table below:

Table 5-1 *Summary of results*

Hypotheses		**Results**
Time efficiency	***H1.*** Users of e-science databases will spend longer time reading learning materials when using query examples with sub-goal or principle labels than using examples without these labels.	Not supported
	H2a. Users of e-science databases will be more efficient in transfer of learning when using query examples with sub-goals than users using examples without this information.	Supported
	H2b. Users of e-science databases will be more efficient in transfer of learning when using query examples with principles than users using examples without this information.	Not supported
Effectiveness	***H3.*** Users of e-science databases will have better initial learning performance when using SQL query examples with sub-goal labels compared to users using examples without this type of information.	Supported
	H4. Users of e-science databases will have better transfer performance when using SQL query examples with principle labels compared to users using examples without this type of information.	Not supported
Possible interaction effects	***H5***. Because of the interaction effect, users of e-science databases will have better initial learning performance when using SQL query examples with sub-goal labels and principles compared to users using examples without two types of information.	Not supported
	H6. Because of the interaction effect, users of e-science databases will have better transfer performance when using SQL query examples with sub-goal labels and principles compared to users using examples without two types of information.	Not supported

The findings of this study are in consistent with some of the prior studies and two hypotheses, ***H2a*** and ***H3***, are supported. The analysis regarding sub-goals proves that giving users examples with specific sub-goals can assist them in applying their learning gains to isomorphic tasks (Catrambone & Holyoak, 1990) and in solving novel problems more efficiently (Morrison, Margulieux & Guzdial, 2015). Sub-goal users did not improve their performance but only increased efficiency in transfer tasks, which may be because the right SQL statements required by question items might not be

explicitly described. For example, participants learnt about ‘sort the results based on program’ in the instructions, but they encountered ‘ensure distinct source’s measurements appear sequentially’ later in their task. By only applying surface similarity between examples learned and novel tasks, participants were not likely to outperform other groups without the process of generalization. This is largely similar to the analogy process of example-based learning as proposed by Anderson, Fincham and Douglass (1997), so that the effectiveness of sub-goals on performance is limited to the initial stage. However, receiving sub-goal information might have increase their confidence so that they completed the transfer tasks faster than non-sub-goal groups.

As for the effect of principles, this study also supports some previous findings (Ummelen,1997; Karreman, 2004) that general instructions, or in other words system information, are less likely to contribute to user performance even combined with certain examples or applied in the context of e-science, in which principles are perceived as useful. One reason behind this finding can be that participants still tend to follow more detailed instructions, both examples and sub-goals, provided at the same time rather than general principles, as proposed by Duff and Barnard (1990). It can be concluded that the usefulness of principles is determined by the certain context they are applied in and their combination with other information. When designing brief and general instructions, principles are likely to improve user’s understanding of procedures and the entire system. On the other hand, when presented together with specific examples, users tend to use information that shares superficial similarity with the task and not to invest much effort in processing the principles.

Regarding time efficiency, ***H1*** is not supported. Participants in each condition invested almost the same amount of time in learning about the examples. Though in most cases additional information will require longer time for reading, this experimental result may be due to easier comprehension when participants were provided with explanations of query code, so that time spent in comprehending and reading compensate for each other as described by Catrambone (1990).

5.2. Implications for practice and suggestions for future research

This study examined the effects of sub-goals and principles on using query examples in terms of

effectiveness and efficiency. The findings further support previous research about the usefulness of principles and sub-goals in the area of procedural instructions. From the perspective of general practice, an implication is that examples are more effective when aided by additional information or in-line notes. Using sub-goal or sub-task descriptions is suggested to break up high-complexity tasks and keep information specificity consistent between examples and other labels, especially in the context of e-science.

In instructional materials, the principle or system information is seldom proven effective for initial task performance. However, according to participants feedback through e-mail after the experiment, group A that received no additional information within examples seemed to have a higher level of information needs. Users of condition A perceived the learning materials as more difficult and less helpful, and a few of them asked the researcher about the correct answer of the tasks. This can be an evidence that users feel uncertainty when facing examples without any further explanation, which is quite common in real practices. Differently, other groups (with additional labels) only reported an eagerness to learn about more examples and even more detailed notes. Although this was not a formal and scientific comparison, we can notice a possibility that when offering instructions for this specific user group, abundant information will result in a higher level of satisfaction and appreciation for the materials. In other words, principles and system information still have a role in e-science applications despite their limited effect in helping users learn and fulfill tasks.

For future research, there is some potential to improve relevant studies. There remains a disparity in empirical findings about principles' effectiveness, while the diversity in research methods and context may have an impact. Eiriksdottir (2011) has pointed that having no effect could result from principles being either ineffectiveness or ignored. Though it is worth considering how researchers interpret their results, in real life instructions there is hardly a way to guarantee that system information is actively processed. Ummelen (1997) and Karreman (2004) have used the click-and-read method to measure how much time users invest in reading declarative information. Eiriksdottir (2011) additionally applied an active study method and asked participants to summarize the principles' main ideas to examine how much they learned. Though the latter approach proved the

usefulness of principles, it is far from the actual fact that users of technical instructions mostly only engage in passive activities such as reading when learning with document-based materials. In further research, the research method should be carefully chosen so that it can not only measure how much users are engaged in processing principles but also whether the procedures of performing tasks are the same as in real use. Besides, other effects such as cognitive load or attitudes are still to be explored, as the particular information needs of e-science users should be considered.

5.3. Limitations

The experiment process may have some limitations that could influence the results. The experiment was conducted completely online, so if the participant was having problem with how to operate the survey, the communication between participants and the researcher was not as efficient as face-to-face. And there was one among 93 students reported difficulty in highlighting the SQL clause in knowledge test. He was given more detailed instructions through email and finished the survey afterwards. The impact of operational issues was minimized by asking participants to complete the knowledge test which contained similar single-choice questions and highlight tasks as in the survey. But there are still some outliers in results caused by technical failures, such as closing the survey window then stop the timer embedded in the survey by mistake. This kind of problem was not detected in time by the researcher when participants were doing the tasks independently.

5.4. Conclusion

As an important instruction type, examples provide users with specific instantiation and allow them to reach pedagogical goals such as learning and transfer of learning. Different from prior research, this study was designed to investigate examples' effect on documentation users rather than on students in classroom settings. An experiment was designed to examine how general principles, and specific sub-goals assisted e-science users in performing SQL tasks. Though it was suggested by other researchers that general principles are effective in generalizing learning gains and specific sub-goals can aid users' initial performance in similar learning tasks, this study has only found

differences between examples with and without sub-goal labels. Sub-goal users showed better performance in learning tasks and higher efficiency in transfer tasks, which confirm previous findings in the e-science context. Besides, principles did not show significant impact on task performance but still seemed to increase users' interest and appreciation. It can be concluded that only specific instructions are proven effective in using examples and should receive a higher priority than general principles when combining additional information with examples in e-science applications.

References

Anderson, J. R., & Fincham, J. M. (1994). Acquisition of procedural skills from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*(6), 1322.

Anderson, J. R., Fincham, J. M., & Douglass, S. (1997). The role of examples and rules in the acquisition of a cognitive skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(4), 932.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, *70*(2), 181-214.

Catrambone, R. (1990). Specific versus general procedures in instructions. *Human-Computer Interaction*, 5, 49–93.

Catrambone, R., & Holyoak, K. J. (1990). Learning subgoals and methods for solving probability problems. *Memory & Cognition*, *18*(6), 593-603.

Catrambone, R. (1995). Following instructions: Effects of principles and examples. *Journal of Experimental Psychology: Applied*, *1*(3), 227.

Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General*, *127*(4), 355.

Chen, X., Mitrovic, A. T., & Matthews, M. (2019). Learning from Worked Examples, Erroneous Examples and Problem Solving: Towards Adaptive Selection of Learning Activities. *IEEE Transactions on Learning Technologies*.

Chi, M., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, 121-152.

Chi, M., Glaser, R., & Rees, E. (1982). Expertise in problem solving. In R. Sternberg (Ed.), *Advances in the Psychology of Human Intelligence (pp. 7-75)*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Chunpir, H., Ludwig, T., & Curri, E. (2014, October). Improving processes for user support in e-Science. In *2014 IEEE 10th International Conference on e-Science* (Vol. 2, pp. 87-90). IEEE.

Chunpir, H. I. (2018, July). How to include users in the design and development of

cyberinfrastructures?. In *International Conference of Design, User Experience, and Usability* (pp. 658-672). Springer, Cham.

Chunpir, H. I., & Ludwig, T. (2014). Reviewing the governance structure of end-user support in e-Science infrastructures. *Informatik 2014*.

Duff, S. C., & Barnard, P. J. (1990, August). Influencing behaviour via device representation; decreasing performance by increasing instruction. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction* (pp. 61-72). North-Holland Publishing Co..

Earle, R. H., Rosso, M. A., & Alexander, K. E. (2015, July). User preferences of software documentation genres. In *Proceedings of the 33rd Annual International Conference on the Design of Communication* (p. 46). ACM.

Eiriksdottir, E., & Catrambone, R. (2011). Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors*, *53*(6), 749-770.

Eiriksdottir, E. (2011). *The role of principles in instructions for procedural tasks: timing of use, method of study, and procedural instruction specificity* (Doctoral dissertation, Georgia Institute of Technology).

Hey, T., & Trefethen, A. E. (2005). Cyberinfrastructure for e-Science. *Science*, *308*(5723), 817-821.

Howe, B., Cole, G., Khoussainova, N., & Battle, L. (2011, June). Automatic example queries for ad hoc databases. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (pp. 1319-1322). ACM.

Huang, Y. H., Lin, K. C., Yu, X., & Hung, J. C. (2015). Comparison of different approaches on example-based learning for novice and proficient learners. *Human-centric Computing and Information Sciences*, *5*(1), 29.

Huang, X. (2017). Example-based learning: Effects of different types of examples on student performance, cognitive load and self-efficacy in a statistical learning task. *Interactive Learning Environments*, *25*(3), 283-294.

Ivanova, M., Nes, N., Goncalves, R., & Kersten, M. (2007, July). Monetdb/sql meets skyserver: the challenges of a scientific database. In *19th International Conference on Scientific and Statistical Database Management (SSDBM 2007)* (pp. 13-13). IEEE.

Karreman, J. (2004). *Use and effect of declarative information in user instructions* (Vol. 18). Rodopi.

Karreman, J., & Steehouder, M. (2004). Some effects of system information in instructions for use. *IEEE transactions on Professional Communication*, *47*(1), 34-43.

Karreman, J., Ummelen, N., & Steehouder, M. (2005, July). Procedural and declarative information in user instructions: What we do and don't know about these information types. In *IPCC 2005. Proceedings. International Professional Communication Conference, 2005.* (pp. 328-333). IEEE.

Keller, J., & Suzuki, K. (2004). Learner motivation and e-learning design: A multinationally validated process. *Journal of Educational Media*, *29*(3), 229-239.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science*, *8*(3), 255-273.

LeFevre, J. A., & Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction*, *3*(1), 1-30.

Li, N., & Thakar, A. R. (2008). CasJobs and MyDB: A batch query workbench. *Computing in Science & Engineering*, *10*(1), 18-29.

Logan, G. D. (1988). Toward and instance theory of automatization. *Psychological Review*, 95, 492–527.

Makiyama, V. H., Raddick, J., & Santos, R. D. (2015, September). Text Mining Applied to SQL Queries: A Case Study for the SDSS SkyServer. In *SIMBig* (pp. 66-72).

Mora, A., González-Núñez, J., Baines, D., Durán, J., Gutiérrez-Sanchéz, R., Racero, E., ... & Segovia, J. C. (2017). The Gaia Archive. *Proceedings of the International Astronomical Union*, *12*(S330), 35-38.

Morrison, B. B., Margulieux, L. E., & Guzdial, M. (2015, July). Subgoals, context, and worked examples in learning computing problem solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 21-29). ACM.

Paas, F. G., & Van Merriënboer, J. J. (1994). Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. *Journal of Educational Psychology*, *86*(1), 122-133.

Reder, L. M., Charney, D. H., & Morgan, K. I. (1986). The role of elaborations in learning a skill

from an instructional text. *Memory & Cognition*, 14(1), 64-78.

Renkl, A. and Atkinson, R.K. (2002). Learning from examples: Fostering self-explanations in computer-based learning environments. *Interactive Learning Environments.* 10, 2 (2002), 105–119.

Renkl, A., Atkinson, R. K., Maier, U. H., & Staley, R. (2002). From example study to problem solving: Smooth transitions help learning. *The Journal of Experimental Education*, *70*(4), 293-315.

Smith, E. E., & Goodman, L. (1984). Understanding written instructions: The role of an explanatory schema. *Cognition and Instruction*, *1*(4), 359-396.

Swarts, J. (2019). Open-source software in the sciences: The challenge of user support. *Journal of Business and Technical Communication*, *33*(1), 60-90.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2, 59-89.

Tomasi, M. D., & Mehlenbacher, B. (1999). Re-engineering online documentation: Designing examples-based online support systems. *Technical communication*, *46*(1), 55-56.

Tsovaltzi, D., McLaren, B. M., Melis, E., & Meyer, A. K. (2013). Erroneous examples: effects on learning fractions in a web-based setting. *International Journal of Technology Enhanced Learning*, *4*(3-4), 191-230.

Ummelen, N. (1997). *Procedural and declarative information in software manuals: Effects on information use, task performance and knowledge* (Vol. 7). Rodopi.

Vallerand, R. J., Pelletier, L. G., Blais, M. R., Briere, N. M., Senecal, C., & Vallieres, E. F. (1992). The Academic Motivation Scale: A measure of intrinsic, extrinsic, and amotivation in education. *Educational and Psychological Measurement*, *52*(4), 1003-1017.

Van der Meij, H. (1995). Principles and heuristics for designing minimalist instruction. *Technical Communication*, *42*(2), 243-261.

Van Der Meij, H. (2003). Minimalism revisited. *Document Design*, 4, 212–233.

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and Instruction*, 7, 1-39.

Zolaktaf, Z. (2017). Facilitating User Interaction With Data. In *PhD@ VLDB*.

Appendix

Appendix 1 Current practices of SQL instructions in e-science databases

As a basic skill in data science, structural query language (SQL) is learnt by students and scientists through books, videos or tutorials. In the case of e-science databases, a scientific query can be complex and difficult to start with a number of unknown table structure and data fields to memorize. As these large databases develop, a focus on user experience prompts engineers and database administrator to provide them with various support for learning and using SQL. Six large e-science database from the disciplinary of astronomy and geo science are selected to give an overview of the current practices of web-based SQL instructions. Types of documentation are observed and sample queries, the examples written in SQL, were particularly focused and information included in each database is listed in the table below. Each information type is explained with selected examples from 6 databases.

Database	SQL Documentation type	Information type in examples
SkyServer (Sloan digital sky survey)	tutorial; sample query; query limits; searching advice	final goal description; table description; syntax explanation; utilization information; sub-goal description; system information; query code
PanSTARRS	how-to guide; FAQ; sample query;	goal description; table used; query code; results
Gaia	tutorials; sample query; timeout notes;	goal; use case; query code; notes; contributions in published articles;
GRASS GIS	notes; sample query; reference manual	goal; notes; query code; sub-goal; utilization information;
PostGIS	tutorial (theory, conceptual knowledge, query templates and query samples)	goal; definition of table used; query code; results/ description of results; system information
LAMOST	query samples in SQL seach interface	goal; query code

Goal description: the aim of using the query

Query the redshift of galaxies during given time. (LAMOST)

We want to get all objects with R degrees of a given position that are high fidelity stellar-like objects. (PanSTARRS)

Table description:

Tables used

- *MeanObjectView*
- *StackObjectAttributes*

(PanSTARRS)

Use case: the situation of using the query

I want to carry out a positional cross-match between two catalogues. (Gaia)

Notes: explanations of code or SQL syntax

Note only the first 10 objects are used to make the query fast. Remove the "top 10" statement to retrieve the full diagram. (Gaia)

The BETWEEN statement can be used to set constraints on a range of values. (Skyserver)

Utilization information: declarative information about the circumstances, time and reasons of using the function

You should use specPhoto whenever your variables of interest can be found there; using specPhoto means that your queries will return results much faster than using JOIN...ON. (Skyserver)

(This query) must be run from the mapset which contains the table. (GRASS)

System information: declarative information about the internal working of the system (Karreman, Ummelen and Steehouder, 2005)

There are several built-in functions available to CAS users that make spatial queries, i.e., those with coordinate cuts, much more efficient than simply including the coordinate constraints in the WHERE clause. (Skyserver)

Query code: SQL expressions that can be copied and modified by users.

```
SELECT gid, name, ST_Area(the_geom) AS area
FROM bc_municipality
WHERE ST_NRings(the_geom) > 1
ORDER BY area DESC LIMIT 1;
```

(PostGIS)

Sub-goal description (written as in-code comments): the action step in each line or in each block

```
SELECT TOP 100
    objID, ra ,dec              -- Get the unique object ID and coordinates
FROM
    PhotoPrimary                -- From the table containing photometric data for unique objects
WHERE
    ra > 185 and ra < 185.1
    AND dec > 15 and dec < 15.1     -- that matches our criteria
```

(Skyserver)

Sub-goal description (stated above code): the action step in each line or in each block

We get all objects within 0.2 degree of RA=334.0 and Dec=0.0 which have mean magnitudes in griz (i.e. at least 1 detection in each band that can be used for the mean mag). In addition, we require

QfPerfect>0.85 in all bands. We select stars with the difference between Kron and PSF magnitude as described here. (PanSTARRS)

Results: data retrieved by the sample query.

```
SELECT
   name,
   ST_Area(the_geom)/10000 AS hectares
FROM
   bc_municipality
ORDER BY hectares DESC
LIMIT 1;

name                | hectares
--------------+-----------------
TUMBLER RIDGE   | 155020.02556131
(1 row)
```

(PostGIS, query code and result)

Contributions in published articles: related researches that have used the query.

DR1. Gaia Collaboration, Brown et al. 2016 A&A 595A, 2G Fig. 4 (Gaia)

Appendix 2 Learning materials

2.1 General introduction

Welcome! SQL is the Structured Query Language, a language widely used in handling data stored in database management systems. Various tasks such as updating and retrieving data can be achieved with SQL statements. And it has been a powerful tool to query sky survey data in astronomy research. This guide provides a brief overview of SQL and query examples are available with comments.

欢迎你学习使用巡天数据库中的 SQL 语言！SQL 即结构化查询语言，是与数据库通信的标准方法。SQL 语句用于执行更新数据库中的数据或从数据库检索数据等任务。在天文学研究中，它被广泛应用于巡天数据的查询和操作。本手册包含了 SQL 的简要介绍，你还可以学习一些包含注释的具体查询示例。

SQL Basics

The data in the database is stored in **tables**, which consist of **columns** and **rows**. In SQL you write **queries** to the database for your tasks. A query consists of the table columns you want to retrieve (the SELECT clause), the table or tables that store the data (the FROM clause) and the conditions to restrict the data you will obtain (the WHERE clause).

数据库中的数据在**表（table）** 中按照**行(row)** 和**列(column)** 组织起来。在 SQL 中，可以通过**查询语句（query）**从数据库中获取数据。查询语句包括要检索的表和列（SELECT 子句）、存储数据的表（FROM 子句）以及查询数据的限制条件（WHERE 子句）。

E.g.
SELECT <columns> FROM <tables> WHERE <conditions>

Mathematical and logical operators
数学与逻辑运算符

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to
+	Add
-	Subtract
*	Multiply

The AND operator displays a record if all conditions separated by AND are TRUE.
The OR operator displays a record if any of conditions separated by OR is TRUE.
The NOT operator displays a record if the condition(s) is NOT TRUE.

SQL Comments
注释
Comments are used to explain sections of SQL statements, or to prevent execution of SQL statements.
Single line comments start with – .Any content between – and the end of the line will be ignored (will not be executed).
注释用于解释 SQL 语句的各个部分，或使 SQL 语句不能运行。
单行注释以–开始，在“–”和行尾中间的所有内容都不会被运行。
E.g.
--Select all:
SELECT * FROM Stars;

Database structure/ table descriptions 数据库结构
Below is a list of the most commonly used tables (and views) in our sky survey database and a short description of them. You can click on the table names to read further information. 以下是该巡天数据库中常用的表和视图，以及对它们的简要介绍。你可以点击表名的链接阅读更多信息。
PhotoObj：stores information about the images of every object, including run, rerun, camcol, field,

ra, dec, magnitudes and object flags. 存储每个目标图像的信息，包括 run，rerun，camcol，field，ra，dec，magnitude 和 object flags 等字段。

PlateX： stores information on the aluminum plates that the sky survey uses to take spectra, including their exposure times and reddening information. You will need to find the Plate and MJD in this table to look up an object's spectrum in the Get Spectra tool. 存储巡天中用于拍摄光谱的铝板信息，还包括曝光时间和红化信息。如需使用 Get Spectra 工具查找目标天体的光谱，你需要在此表中找到 Plate 和 MJD 两个字段的值。

SpecObj ： stores information on objects' spectra, including redshifts and spectroscopic classifications. 存储有关目标天体光谱的信息，包括红移和光谱分类。

Neighbors：stores all PhotoObj pairs within 30 arcseconds. 存储所有距离在 30 角秒（arcsecond）内的 PhotoObj 对。

Photoz：stores the photometrically estimated redshifts for all objects in the GalaxyTag view. 存储 GalaxyTag 视图中所有目标的测光红移。

Our sky survey data also contains several subsets of the PhotoObj table. ***PhotoPrimary*** contains only the "best" measurements of each object. Generally, it's better to use PhotoPrimary rather than PhotoObj, which contains both good and bad data. ***Star*** contains only data for stars, ***Galaxy*** contains only data for galaxies, and Unknown contains only data for objects classified as "unknown." These subsets are actually views rather than tables; **you will get familiar with them later in through examples.**

我们的巡天数据还包含 PhotoObj 表的几个子集。 PhotoPrimary 仅包含每个对象的"最佳"测量值。通常，最好使用 PhotoPrimary 而不是 PhotoObj 因为后者的数据质量不一。 Star 仅包含恒星的数据，Galaxy 仅包含星系数据，而 Unknown 仅包含被归类为"未知"对象的数据。这些子集实际上是视图而不是表。**你将通过后面的示例进一步熟悉它们。**

Sky survey basics 巡天数据库常用字段

ra: right ascension (sky longitude) ;赤经

dec: declination (sky latitude) ;赤纬

g magnitude: model magnitude in g filter ;g 波段的星等

2.1 Examples in condition A

Example 1

– Find unique objects in an RA/Dec box. 查询一个赤经/赤纬坐标区域内具有唯一性的目标。

–Table used: PhotoObj (PhotoPrimary is a view created based on PhotoObj)

```
SELECT TOP 100 objID,
  field, ra, dec
FROM PhotoPrimary
WHERE
  ra > 185. and ra < 185.1 and dec > 15. and dec < 15.1
```

Example 2

– Find all galaxies in a certain area of the sky that meet specific criteria in their measured parameters. 查询天区中参数符合特定判据的所有星系。

– Table used: Galaxy

```
SELECT TOP 100 objID, ra, dec, cModelMag_g
FROM Galaxy
WHERE ra BETWEEN 178 AND 180
  AND dec < 0
  AND cModelMag_g BETWEEN 18 AND 19
```

Example 3

–Look for spectra of quasars and the date and time at which each spectrum was taken. 查询类星体的光谱，并显示每个光谱的拍摄日期和时间。

– Table used: SpecPhoto (a view which is a pre-computed join of the most commonly-searched fields in both the SpecObj view that contains spectroscopy data and the PhotoObj view that contains photometry data)

```
SELECT TOP 100
  sp.objID, sp.ra, sp.dec,
  px.taiBegin, px.taiEnd,
FROM specPhoto AS sp
JOIN plateX AS px
ON sp.plateID = px.plateID
WHERE
  (sp.class='QSO')
```

Example 4

– Count the number of spectra of each spectral classification (galaxy, quasar, star). 查询每种光谱分类（星系，类星体，恒星）的光谱数。

– Table used: SpecObj

```
SELECT class, count(*)
FROM SpecObj
GROUP BY class
```

Example 5

– Find all objects within 30 arcseconds of one another that have very similar colors.查询彼此角距离小于 30 角秒且颜色相近的所有目标。

– Table used: PhotoPrimary, Neighbors

```
SELECT TOP 10 P.ObjID
FROM PhotoPrimary AS P
  JOIN Neighbors AS N ON P.ObjID = N.ObjID
  JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID
WHERE
  P.ObjID < L. ObjID
  and abs((P.u-P.g)-(L.u-L.g))<0.05
  and abs((P.g-P.r)-(L.g-L.r))<0.05
  and abs((P.r-P.i)-(L.r-L.i))<0.05
  and abs((P.i-P.z)-(L.i-L.z))<0.05
```

Example 6

– List the number of each type of object observed by each special spectroscopic observation program. 查询每个光谱观测项目观测到的每种目标的数量。

– Table used: SpecObjAll, PlateX

```
SELECT plate.programname, class,
  COUNT(specObjId) AS numObjs
  FROM SpecObjAll
  JOIN PlateX AS plate ON plate.plate = specObjAll.plate
  GROUP BY plate.programname, class
  ORDER BY plate.programname, class
```

2.2 Examples in condition B

Example 1

– Find unique objects in an RA/Dec box. 查询一个赤经/赤纬坐标区域内具有唯一性的目标。

–Table used: PhotoObj (PhotoPrimary is a view created based on PhotoObj)

```
SELECT TOP 100 objID,      -- The SELECT TOP clause is used to specify the number of records to return.
  field, ra, dec
FROM PhotoPrimary          -- Selecting from a view which contains only unique objects can avoid duplicates in the result.
WHERE
  ra > 185. and ra < 185.1 and dec > 15. and dec < 15.1
```

Example 2

– Find all galaxies in a certain area of the sky that meet specific criteria in their measured parameters. 查询天区中参数符合特定判据的所有星系。

– Table used: Galaxy

```
SELECT TOP 100 objID, ra, dec, cModelMag_g      --The cModelMag_g column contains “model  magnitude in g filter” .
FROM Galaxy
WHERE ra BETWEEN 178 AND 180                    --The BETWEEN operator selects values within a given range.
  AND dec < 0
  AND cModelMag_g BETWEEN 18 AND 19
```

Example 3

– Look for spectra of quasars and shows the date and time at which each spectrum was taken. 查询类星体的光谱，并显示每个光谱的拍摄日期和时间。

– Table used: SpecPhoto (a view which is a pre-computed join of the most commonly-searched fields in both the SpecObj view that contains spectroscopy data and the PhotoObj view that contains photometry data)

```
SELECT TOP 100
```

```
   sp.objID, sp.ra, sp.dec,
   px.taiBegin, px.taiEnd,
-- Aliases (names) are used to give a table, or a column a temporary name.
FROM specPhoto AS sp
-- A JOIN clause is used to combine rows from two or more tables, based on key variab
le they have in common.
JOIN plateX AS px
ON sp.plateID = px.plateID
WHERE
   (sp.class='QSO')
```

Example 4

– Count the number of spectra of each spectral classification (galaxy, quasar, star). 查询每种光谱分类（星系，类星体，恒星）的光谱数。

– Table used: SpecObj

```
-- The count(*) statement returns the number of all records that meet specific search criter
ia
SELECT class, count(*)
FROM SpecObj
-- The GROUP BY statement groups results into groups (categories) based on the value o
f a data column.
GROUP BY class
```

Example 5

– Find all objects within 30 arcseconds of one another that have very similar colors.查询彼此角距离小于 30 角秒且颜色相近的所有目标。

– Table used: PhotoPrimary, Neighbors

```
SELECT TOP 10 P.ObjID
FROM PhotoPrimary AS P
   JOIN Neighbors AS N ON P.ObjID = N.ObjID
   JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID
WHERE
   P.ObjID < L. ObjID
-- If two objects' color ratios u-g, g-r, r-i and i-z are less than 0.05m, they have a simila
r spectra.
-- The abs() function returns the absolute value of a number.

   and abs((P.u-P.g)-(L.u-L.g))<0.05
   and abs((P.g-P.r)-(L.g-L.r))<0.05
   and abs((P.r-P.i)-(L.r-L.i))<0.05
   and abs((P.i-P.z)-(L.i-L.z))<0.05
```

Example 6

– List the number of each type of object observed by each special spectroscopic observation

program. 查询每个光谱观测项目观测到的每种目标的数量。
– Table used: SpecObjAll, PlateX

```
SELECT plate.programname, class,
-- The COUNT() function returns the number of rows that matches a specified criteria.
  COUNT(specObjId) AS numObjs
  FROM SpecObjAll
  JOIN PlateX AS plate ON plate.plate = specObjAll.plate
-- The GROUP BY statement sorts results into groups (categories) based on the value of a data column.
  GROUP BY plate.programname, class
-- The ORDER BY statement is used to sort the result-set in ascending or descending order.
  ORDER BY plate.programname, class
```

2.3 Examples in condition C

Example 1
– Find unique objects in an RA/Dec box. 查询一个赤经/赤纬坐标区域内具有唯一性的目标。
–Table used: PhotoObj (PhotoPrimary is a view created based on PhotoObj)

```
SELECT TOP 100 objID,          -- to retrieve only the first 100 object ID
  field, ra, dec               -- to get the field number, and coordinates
FROM PhotoPrimary              -- to select from PhotoPrimary which contains only unique objects in PhotoObj
WHERE
  ra > 185. and ra < 185.1 and dec > 15. and dec < 15.1
```

Example 2
– Find all galaxies in a certain area of the sky that meet specific criteria in their measured parameters. 查询天区中参数符合特定判据的所有星系。
– Table used: Galaxy

```
SELECT TOP 100 objID, ra, dec, cModelMag_g  -- to get first 100 objectID, coordinates and g magnitude
FROM Galaxy
WHERE ra BETWEEN 178 AND 180                -- to select objects of which right ascension ranges between 178 and 180
  AND dec < 0
  AND cModelMag_g BETWEEN 18 AND 19         -- to select objects of which g magnitude ranges between 18 and 19
```

Example 3

– This query looks for spectra of quasars and shows the date and time at which each spectrum was taken. 查询类星体的光谱，并显示每个光谱的拍摄日期和时间。

– Table used: SpecPhoto (a view which is a pre-computed join of the most commonly-searched fields in both the SpecObj view that contains spectroscopy data and the PhotoObj view that contains photometry data)

```
SELECT TOP 100
  sp.objID, sp.ra, sp.dec,  -- to get first 100 objectID and coordinates
  px.taiBegin, px.taiEnd,   -- to select beginning time and ending time
FROM specPhoto AS sp        -- to give specPhoto an alias (nickname) as sp in this qu
ery
JOIN plateX AS px           -- to give plateX an alias (nickname)as px in this query
ON sp.plateID = px.plateID  -- to use JOIN…ON… to select objects that stored in both
sp and px
WHERE
  (sp.class='QSO')          -- to select QSO from spectroscopic class (GALAXY, QSO,
or STAR)
```

Example 4

– This query counts the number of spectra of each spectral classification (galaxy, quasar, star). 查询每种光谱分类（星系，类星体，恒星）的光谱数。

– Table used: SpecObj

```
SELECT class, count(*)   -- to retrieve class and the total number of all records
FROM SpecObj
-- to group the number of records based on 'class' column which contains the spectral cla
ssification of the object
GROUP BY class
```

Example 5

– This query finds all objects within 30 arcseconds of one another that have very similar colors.查询彼此角距离小于 30 角秒且颜色相近的所有目标。

– Table used: PhotoPrimary, Neighbors

```
SELECT TOP 10 P.ObjID
FROM PhotoPrimary AS P      -- to select primary objects from P
  JOIN Neighbors AS N ON P.ObjID = N.ObjID
  JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID  -- to select lens candidate
from L
WHERE
  P.ObjID < L. ObjID        -- to avoid duplicates
-- to select objects from L and P that have similar spectra, which means the color ratios
u-g, g-r, r-i and i-z are less than 0.05m
  and abs((P.u-P.g)-(L.u-L.g))<0.05
  and abs((P.g-P.r)-(L.g-L.r))<0.05
```

```
  and abs((P.r-P.i)-(L.r-L.i))<0.05
  and abs((P.i-P.z)-(L.i-L.z))<0.05
```

Example 6

– This query lists the number of each type of object observed by each special spectroscopic observation program. 查询每个光谱观测项目观测到的每种目标的数量。

– Table used: SpecObjAll, PlateX

```
SELECT plate.programname, class,
  COUNT(specObjId) AS numObjs                    --to count objects and name the result column as numObjs
  FROM SpecObjAll
  JOIN PlateX AS plate ON plate.plate = specObjAll.plate
  GROUP BY plate.programname, class              -- to categorize results based on program and class
  ORDER BY plate.programname, class              --to sort the results based on program and in each program sort results based on class
```

2.4 Examples in condition D

Example 1

– Find unique objects in an RA/Dec box. 查询一个赤经/赤纬坐标区域内具有唯一性的目标。

–Table used: PhotoObj (PhotoPrimary is a view created based on PhotoObj)

```
-- to retrieve only the first 100 object ID
-- The SELECT TOP clause is used to specify the number of records to return.
SELECT TOP 100 objID,
  field, ra, dec            -- to get the field number and coordinates
-- to select from PhotoPrimary which contains only unique objects in PhotoObj
-- Selecting from a view which contains only unique objects can avoid duplicates in the result.
FROM PhotoPrimary
WHERE
  ra > 185. and ra < 185.1 and dec > 15. and dec < 15.1
```

Example 2

– Find all galaxies in a certain area of the sky that meet specific criteria in their measured parameters. 查询天区中参数符合特定判据的所有星系。

– Table used: Galaxy

```
--to get first 100 objectID, coordinates and g magnitude
--The cModelMag_g column contains “model magnitude in g filter”.
SELECT TOP 100 objID, ra, dec, cModelMag_g
FROM Galaxy
--to select objects of which right ascension ranges between 178 and 180
```

```
--The BETWEEN operator selects values within a given range.
WHERE ra BETWEEN 178 AND 180
  AND dec < 0
--to select objects of which g magnitude ranges between 18 and 19
  AND cModelMag_g BETWEEN 18 AND 19
```

Example 3

–Look for spectra of quasars and the date and time at which each spectrum was taken. 查询类星体的光谱，并显示每个光谱的拍摄日期和时间。

– Table used: SpecPhoto (a view which is a pre-computed join of the most commonly-searched fields in both the SpecObj view that contains spectroscopy data and the PhotoObj view that contains photometry data)

```
SELECT TOP 100
  sp.objID, sp.ra, sp.dec,    -- to get first 100 objectID and coordinates
  px.taiBegin, px.taiEnd,     -- to select beginning time and ending time
-- Aliases (names) are used to give a table, or a column a temporary name.
FROM specPhoto AS sp          -- to give specPhoto an alias (nickname) as sp in this query
-- to use JOIN…ON… to select objects that stored in both sp and px
-- A JOIN clause is used to combine rows from two or more tables, based on key variable they have in common.
JOIN plateX AS px             -- to give plateX an alias (nickname)as px in this query
ON sp.plateID = px.plateID
WHERE
  (sp.class='QSO')            -- to select QSO from spectroscopic class (GALAXY, QSO, or STAR)
```

Example 4

– Count the number of spectra of each spectral classification (galaxy, quasar, star). 查询每种光谱分类（星系，类星体，恒星）的光谱数。

– Table used: SpecObj

```
-- to retrieve class and the total number of records
-- the count(*) statement returns the number of all records that meet specific search criteria
SELECT class, count(*)
FROM SpecObj
-- to group the number of records based on 'class' column which contains the spectral classification of the object
-- The GROUP BY statement groups results into groups (categories) based on the value of a data column.
GROUP BY class
```

Example 5

– Find all objects within 30 arcseconds of one another that have very similar colors.查询彼此角距

离小于 30 角秒且颜色相近的所有目标。
– Table used: PhotoPrimary, Neighbors

```
SELECT TOP 10 P.ObjID
FROM   PhotoPrimary AS P                                    -- to select primary objects from
 P
  JOIN Neighbors AS N ON P.ObjID = N.ObjID
  JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID    -- to select lens candidate
 from L
WHERE
  P.ObjID < L. ObjID                                          -- to avoid duplicates
-- to select objects from L and P that have similar spectra, which means the color ratios
u-g, g-r, r-i are less than 0.05m
-- If two objects' color ratios u-g, g-r, r-i are less than 0.05m, they have a similar spectr
a.
  and abs((P.u-P.g)-(L.u-L.g))<0.05
  and abs((P.g-P.r)-(L.g-L.r))<0.05
  and abs((P.r-P.i)-(L.r-L.i))<0.05
  and abs((P.i-P.z)-(L.i-L.z))<0.05          -- The ABS() function returns the absolute value
 of a number.
```

Example 6

– List the number of each type of object observed by each special spectroscopic observation program. 查询每个光谱观测项目观测到的每种目标的数量。
– Table used: SpecObjAll, PlateX

```
SELECT plate.programname, class,
-- to count objects and name the result column as numObjs
-- The COUNT() function returns the number of rows that matches a specified criteria.
  COUNT(specObjId) AS numObjs
  FROM SpecObjAll
  JOIN PlateX AS plate ON plate.plate = specObjAll.plate
-- to categorize results based on program and class
-- The GROUP BY statement sorts results into groups (categories) based on the value of
a data column.
  GROUP BY plate.programname, class
-- to sort the results based on program and in each program sort results based on class
-- The ORDER BY keyword is used to sort the result-set in ascending or descending orde
r.
  ORDER BY plate.programname, class
```

Appendix 3 Knowledge test

(Correct answers are in bold letters.)

1. Which SQL statement is used to extract data from a database?

A. EXTRACT

B. SELECT

C. OPEN

D. GET

2. Which SQL statement is used to insert new data in a database?

A. INSERT NEW

B. ADD RECORD

C. ADD NEW

D. INSERT INTO

3. With SQL, how do you select a column named "FirstName" from a table named "Persons"?

A. SELECT Persons.FirstName

B. SELECT FirstName FROM Persons

C. EXTRACT FirstName FROM Person

D. EXTRACT Person.FirstName

4. With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" is "Peter"?

A. SELECT [all] FROM Persons WHERE FirstName LIKE 'Peter'

B. SELECT [all] FROM Persons WHERE FirstName='Peter'

C. SELECT * FROM Persons WHERE FirstName='Peter'

D. SELECT * FROM Persons WHERE FirstName<>'Peter'

5. With SQL, how can you return all the records from a table named "Persons" sorted descending by "FirstName"?

A. SELECT * FROM Persons SORT 'FirstName' DESC

B. SELECT * FROM Persons ORDER BY FirstName DESC

C. SELECT * FROM Persons ORDER FirstName DESC

D. SELECT * FROM Persons SORT BY 'FirstName' DESC

6. In the following query, could you highlight the clause(s) that give(s) columns or tables a nickname/ alias?

--Compare different redshift measurements of the same object for objects with high redshift

SELECT prim.bestObjId,
prim.mjd AS PrimMJD, prim.plate AS PrimPlate,
other.mjd AS OtherMJD, other.plate AS OtherPlate,
prim.z AS PrimZ, other.z AS OtherZ,
plate.programname
FROM SpecObjAll prim
JOIN SpecObjAll other ON prim.bestObjId = other.bestObjId
JOIN platex AS plate ON other.plate = plate.plate AND other.mjd = plate.mjd

WHERE other.bestObjId > 0
AND prim.sciencePrimary = 1
AND other.sciencePrimary = 0
AND prim.z > 2.5
ORDER BY prim.bestObjId

I don't know

7. In the following query, could you highlight the clause(s) that specify(ies) the condition of combining objects in multiple tables?
-- Find galaxies that are blended with a star, and output the deblended galaxy magnitudes

SELECT TOP 10 G.ObjID, G.u, G.g, G.r, G.i, G.z
FROM Galaxy AS G
JOIN Star AS S
ON G.parentID = S.parentID
WHERE G.parentID > 0

I don't know

8. In the following query, could you highlight the keyword(s) that return(s) the number of objects matching the specific critiria?
-- This query returns a histogram of absolute magnitudes for the reliable redshift estimation in the 0.4 < z < 0.5 range.

SELECT round(absMagR,1) as absMagR,
COUNT(*) as cnt
FROM Photoz
WHERE
z BETWEEN 0.4 and 0.5
AND photoErrorClass=1 and cnt>95
AND zErr BETWEEN 0 and 0.03
group by round(absMagR,1)
order by round(absMagR,1)

I don't know

9. When using SDSS DR15 Image List Tool, all Image List queries must have a clause to retrieve coordinate pairs of objects. In the following query, could you highlight the clause(s) used for this function?

SELECT TOP 100 P1.objID AS name,
P1.ra AS ra, P1.dec AS dec,
FROM PhotoTag P1,

Neighbors N,
PhotoTag P2
WHERE P1.objID = N. objID
AND P2.objID = N.NeighborObjID AND N.Distance < 0.05

I don't know

10. In the following query, could you highlight the clause(s) that ensure(s) that each distinct source's measurements appear sequentially in the results?

SELECT p.*, c.*
FROM gaiadr1.cepheid AS c, gaiadr1.phot_variable_time_series_gfov AS p
WHERE p.source_id = c.source_id
ORDER BY p.source_id, p.observation_time

I don't know

Appendix 4 Survey questions

(Correct answers are in bold letters.)

Exercise 1
Please select the correct answer or highlight the clause(s) as described. In highlight questions, the answer can be one or more clauses. To highlight the code, please click on each line and chose "highlight".

Please don't use the example book when answering this part.

1. With SQL, how do you select first 100 objID from a table/ view named “PhotoPrimary”?
A. SELECT objID FROM PhotoPrimary
B. SELECT TOP 100 PhotoPrimary.objID.
C. EXTRACT objID FROM PhotoPrimary
D. SELECT TOP 100 objID FROM PhotoPrimary
E. I don’t know

2. With SQL, which clause would you use to sort the results?
A. ORDER
B. SORT
C. ORDER BY
D. SORT BY
E. I don’t know

3. With SQL, which clause would you use to select objects that their values are in a given range?

A. IN
B. BETWEEN
C. RANGE
D. JOIN
E. I don’t know

4. With SQL, which clause would you use to categorize results based on the value of one or more column?
A. GROUP BY
B. ORDER BY
C. JOIN
D. SELECT INTO
E. I don’t know

5. With SQL, which clause would you use to select rows in more than one table based on their common column?
A. UNION
B. GROUP BY
C. JOIN
D. COMBINE
E. I don’t know

6. In the following query, could you highlight the clause(s) that give(s) the table a nickname to make it easy to read and write?

SELECT TOP 100
sp.objID, sp.ra, sp.dec,
px.taiBegin, px.taiEnd,
FROM specPhoto AS sp
JOIN plateX AS px
ON sp.plateID = px.plateID
WHERE
(sp.class='QSO')

I don't know

7. In the following query, could you highlight the clause(s) that select(s) objects from a certain range of g magnitude?

SELECT TOP 100 objID, ra, dec, cModelMag_g
FROM Galaxy
WHERE ra BETWEEN 178 AND 180
AND dec < 0
AND cModelMag_g BETWEEN 18 AND 19

I don't know

8. In the following query, could you highlight the clause(s) that specify(ies) the criteria for smilar spectra (colors)?

SELECT TOP 10 P.ObjID
FROM PhotoPrimary AS P
JOIN Neighbors AS N ON P.ObjID = N.ObjID
JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID
WHERE
P.ObjID < L. ObjID
and abs((P.u-P.g)-(L.u-L.g))<0.05
and abs((P.g-P.r)-(L.g-L.r))<0.05
and abs((P.r-P.i)-(L.r-L.i))<0.05
and abs((P.i-P.z)-(L.i-L.z))<0.05

I don't know

9. In the following query, could you highlight the clause(s) that avoid(s) duplicates in results?

SELECT TOP 10 P.ObjID
FROM PhotoPrimary AS P
JOIN Neighbors AS N ON P.ObjID = N.ObjID
JOIN PhotoPrimary AS L ON L.ObjID = N.NeighborObjID
WHERE
P.ObjID < L. ObjID
and abs((P.u-P.g)-(L.u-L.g))<0.05
and abs((P.g-P.r)-(L.g-L.r))<0.05
and abs((P.r-P.i)-(L.r-L.i))<0.05
and abs((P.i-P.z)-(L.i-L.z))<0.0

I don't know

10. In the following query, could you highlight the clause(s) that count(s) objects and names the result column?

SELECT plate.programname, class,
COUNT(specObjId) AS numObjs
FROM SpecObjAll
JOIN PlateX AS plate ON plate.plate = specObjAll.plate
GROUP BY plate.programname, class
ORDER BY plate.programname, class

I don't know

Exercise 2
Please highlight the clause(s) as described. In highlight questions, the answer can be one or more clauses.

You can refer to the example book.

1. In the following query, could you highlight the clause(s) that give(s) columns or tables a nickname/ alias?
--Compare different redshift measurements of the same object for objects with high redshift

```
SELECT prim.bestObjId,
prim.mjd AS PrimMJD, prim.plate AS PrimPlate,
other.mjd AS OtherMJD, other.plate AS OtherPlate,
prim.z AS PrimZ, other.z AS OtherZ,
plate.programname
FROM SpecObjAll prim
JOIN SpecObjAll other ON prim.bestObjId = other.bestObjId
JOIN platex AS plate ON other.plate = plate.plate AND other.mjd = plate.mjd
WHERE other.bestObjId > 0
AND prim.sciencePrimary = 1
AND other.sciencePrimary = 0
AND prim.z > 2.5
ORDER BY prim.bestObjId
```

I don't know

2. In the following query, could you highlight the clause(s) that specify(ies) the condition of combining objects in multiple tables?
-- Find galaxies that are blended with a star, and output the deblended galaxy magnitudes

```
SELECT TOP 10 G.ObjID, G.u, G.g, G.r, G.i, G.z
FROM Galaxy AS G
JOIN Star AS S
ON G.parentID = S.parentID
WHERE G.parentID > 0
```

I don't know

3. In the following query, could you highlight the keyword(s) that return(s) the number of objects matching the specific critiria?
-- This query returns a histogram of absolute magnitudes for the reliable redshift estimation in the

0.4 < z < 0.5 range.

SELECT round(absMagR,1) as absMagR,
COUNT(*) as cnt
FROM Photoz
WHERE
z BETWEEN 0.4 and 0.5
AND photoErrorClass=1 and cnt>95
AND zErr BETWEEN 0 and 0.03
group by round(absMagR,1)
order by round(absMagR,1)

I don't know

4. When using SDSS DR15 Image List Tool, all Image List queries must have a clause to retrieve coordinate pairs of objects. In the following query, could you highlight the clause(s) used for this function?

SELECT TOP 100 P1.objID AS name,
P1.ra AS ra, P1.dec AS dec,
FROM PhotoTag P1,
Neighbors N,
PhotoTag P2
WHERE P1.objID = N. objID
AND P2.objID = N.NeighborObjID AND N.Distance < 0.05

I don't know

5. In the following query, could you highlight the clause(s) that ensure(s) that each distinct source's measurements appear sequentially in the results?

SELECT p.*, c.*
FROM gaiadr1.cepheid AS c, gaiadr1.phot_variable_time_series_gfov AS p
WHERE p.source_id = c.source_id
ORDER BY p.source_id, p.observation_time

I don't know