Understandability of Generated Database Query Descriptions

Adam Johansson, (adajoh@kth.se)
Hugo Hägglund, (hugoha@kth.se)

May 12, 2014
ABSTRACT

By creating a relational database describing scenes in pictures and lightly configuring the C-Phrase natural language interface on top of this database, letting the configuration be of minimal cost with naming of names, relations and attributes and a very small set of lexical rules this paper finds that the generated texts are only marginally less understandable by the general public than handwritten texts.
SAMMANFATTNING

Genom att skapa en relationsdatabas som beskriver scener i bilder och att implementera ett gränssnitt i naturligt språk, C-Phrase, ovanpå en databas, syftar denna uppsats att undersöka försäkelse av genererade texter som beskriver databasfrågor. Konfigurationen av C-Phrase är av minimalistisk typ för att simulera ett mer realistiskt scenario, endast namngivning av relationer, attribut och joins samt en liten mängd lexikala regler. Uppsatsen finner att förståelsen för genererade beskrivningar av databasfrågor i C-Phrase endast är marginellt svårare att förstå än motsvarande beskrivningar skrivna av en människa.
To Michael Minock, our database guru.
# CONTENTS

1 BACKGROUND 7

1 INTRODUCTION 8
  1.1 Preface 8
  1.2 Purpose 8
  1.3 Problem Statement 9

2 DATABASE QUERIES 10
  2.1 SQL 10
  2.2 Tuple Calculus 10

3 NATURAL LANGUAGE GENERATION 11
  3.1 What is NLG? 11
  3.2 Purpose of NLG 11
  3.3 Difficulties of NLG 12
  3.4 C-Phrase 12
  3.5 C-Phrase and NLG 12

4 METHOD 14

4 CONFIGURATION 15
  4.1 Setting up the database 15
  4.2 Configuring C-Phrase 15

5 EVALUATION METHOD 17
  5.1 Problems With Evaluation 17
  5.2 Our Survey Method 17
  5.3 Survey Subjects 19
  5.4 Survey calculations 19

6 RESULTS 20

7 DISCUSSION 25
  7.1 Possible Improvements 25
  7.2 What Went Well 26
  7.3 Conclusions 26

8 APPENDIX 28
  A SCHEMA 29
  B QUERIES 31
  C HANDWRITTEN QUESTIONS 34
  D GENERATED QUESTIONS 35
ACRONYMS

NLG  Natural Language Generator
NLP  Natural Language Processing
NLI  Natural Language Interface
API  Application Programming Interface
GUI  Graphical User Interface
SQL  Structured Query Language
DBMS Database Management System
QBE  Query By Example
Part I

BACKGROUND
INTRODUCTION

1.1 PREFACE

This bachelor thesis was written as part of the course DD143X (dkand13), Degree Project in Computer Science at the school of Computer Science and Communications, Royal Institute of Technology in Stockholm.

The authors are third year students of the computer science programme. The paper was written under the supervision of Michael Minock (minock@kth.se) and the course examiner was Karl Meinke (karlm@csc.kth.se).

The thesis investigates natural language generation of database queries in a minimal configuration of the C-Phrase system and how well these queries are understood by humans compared to a database query rewritten by a human.

1.2 PURPOSE

This project is about evaluating the understanding of natural language generation of database queries in the C-Phrase system. By just lightly naming joins, attributes and relations and not doing extensive ontology configuration one can hopefully show what the C-Phrase capabilities are at the minimum when it comes to regular user understanding. By not overly configuring it one might simulate an administrator setting C-Phrase up with low or no skill and interest to achieve any great results in order to retain realism of our evaluation.

It is the keyword natural in natural language generation that makes this subject interesting and tricky. Most people don’t know how to write or read a database query and are not interested in learning it. With both good natural language processing of queries and good natural language generation of queries a database can become much more accessible to the general public without spending an immense amount of manpower creating understandable texts. Creating such a system is however not very easy, there is a big risk of creating texts that are cryptic and ambiguous.

This project is based on C-Phrase which is a natural language interface for databases. In this system one can enter a query in natural
text and it is interpreted as a database query. The results of the query and a representation of the query submitted is generated in natural language.

1.3 PROBLEM STATEMENT

How understandable can the generated text describing queries be by simply naming joins, attributes and relations and doing some minimal ontology configuration in C-Phrase? How does this compare to someone manually writing this descriptive text? By creating a database describing scenes in pictures and configuring C-Phrase to interpret and generate text from interesting queries run on this database to find the answer to the aforementioned questions.
DATABASE QUERIES

2.1 SQL

SQL is the most commonly used relational DBMS’s query language. SQL serves as both a data-definition and data-manipulating language i.e. it can be used to create, update and fetch a result from a query. The query part is the one that is focused in this section. The query part in SQL has capabilities very close to those of relational algebra, hence it is a powerful query language. The queries are very conditional based, i.e. the WHERE clause that specifies when something is supposed to happen is a very dominant part of the language.\[^1\]

2.2 TUPLE CALCULUS

Tuple calculus has had a big influence in the design of the aforementioned SQL and other relational database query languages such as Query-By-Example(QBE). Tuple calculus was introduced 1970 by Edgar Codd. A so called tuple can be considered a row in a database table. A tuple variable is a variable that is set to a particular relation schema’s tuples in order to evaluate a query.

The form of a tuple query is \{x \mid P(x)\}, where x is a tuple variable and p is the formula that describes x. The result of this query is the set of tuples that the formula \(P(x)\) evaluates to true for. An example query: \{x \mid People(x) \land x.hat = "red"\}. In this example \(P(x) = People(x) \land x.hat = "red"\). This query could be described as all x:s where x is part of the table people and has the attribute hat set equal to red.\[^1\]
3

NATURAL LANGUAGE GENERATION

3.1 WHAT IS NLG?

Natural language generation files under artificial intelligence and computational linguistics and the purpose of NLG is to automatically generate texts such as documents, reports and, which we will investigate in this thesis, descriptions of database queries. Output does not necessarily have to be text, it might as well be speech, but for the purposes of this thesis only text generation will be considered. Summarizing, one can say that NLG is the process of turning computer representations of data into human readable representations. [7]

NLG can be divided into four separate problems, content determination, macroplanning, microplanning and realisation. Content determination is mainly about deciding which facts we want to communicate, macroplanning is the overall structure of the document, order of paragraphs etc. Microplanning is a quite big part that decides how the actual texts should look like, which facts are part of which sentences, which are aggregated etc. Realisation is the part where an actual grammar is applied and the text is put into its final form. [6]

3.2 PURPOSE OF NLG

As stated in the previous section, NLG turns computer representations of data into a more accessible and readable form. The uses vary wildly, from automated weather reports and tools to help write job descriptions to creating software manuals and helping a non-expert user to query a database. NLG is not always the best solution however, sometimes a simple graph does the job a lot better or the benefits of creating an NLG system for a specific purpose is outweighed by the pure economical costs.

An example of the latter can be a text that will not be produced many times, so, in practice it is cheaper to have a human write the text. [7] As E. Reiter also states in [7], a text from 1997, a specific problem might be unsolveable with the current NLG technology. Undoubtedly, progress has been made since 1997 but NLG is still not a fully explored field of research.
3.3 DIFFICULTIES OF NLG

On a more general level, NLG is composed of choice problems. This sets it a bit apart from NLP which has a lot less choice making to do. An example of choice for NLG is that the system must select words so that the text become linguistically correct, i.e. to choose to use reflexive pronouns when needed or to select the plural form of the word.

NLG systems also must select the choice that is best for its target audience, in security instructions for nuclear power plants a very verbose and overdetailed text might be preferred, in another case it might be the time spent reading that is of most importance to the audience and the correct choice in that context is to choose a shorter phrase.

The problem of choices make it much harder to create a general NLG system but an at least partly authoring based solution could alleviate some of these issues. These choice problems are however so big that very few or no NLG systems have consistently been used over a period of years by a big group of users.[6]

3.4 C-PHRAISE

The C-Phrase system is a system that provides a natural language interface to a database. It allows an administrator, with some database and language skills to configure the system by lightly naming joins and relations as well as building a semantic grammar that interprets the queries. C-Phrase include both a natural language processing (NLP) part and natural language generation part.

The system allows a user to enter a query in natural text, interprets that query and translates it to tuple calculus and then to SQL. The SQL query is sent to the database and the result goes through an NLG process and is displayed for the user. C-Phrase also takes the tuple calculus query and again generate natural text from it, allowing a user to see if the system did a correct interpretation. If the system detected ambiguity or misconceptions in the user query it can communicate this to the user allowing the user to refine his query. [5]

3.5 C-PHRAISE AND NLG

The NLP component of C-Phrase uses a slightly modified $\tilde{X}$-theory as the more strategic component, defining the order of nounphrases and it’s PRE and POST modifiers. $\lambda$-SCFG, (semantic context free grammar) is used to define lexical rules, for every named relation the system generates a few default rules, these rules are modifiable in the interface under the Onthology tab, but can also be defined directly with a text editor which can sometimes come in handy.[8]
The \( \lambda \)-SCFG *sentence pattern* rules define the mapping of a query to text and if it is a PRE, HEAD or POST element. These elements can also be configured with a number of other attributes, the most important being \textit{silent} and \textit{proper} attributes. [3] [2]

If the \textit{silent} attribute is chosen, the system will match on the provided text when processing natural language, but it will not generate this text when processing queries and generating natural language. The \textit{proper} attribute allows the administrator to set a HEAD elements text as a proper noun. For example it can be used to allow the type of object to act as a real reference to the object. Instead of the user having to enter \textit{Objects of type chair} you can then just ask the system for \textit{Chairs}. [5]

The NLG component, filling more of a feedback function, doesn’t use the \( \tilde{X} \)-theory, it simple matches on the \( \lambda \)-SCFG *sentence pattern* rule and then randomly choosing PRE, HEAD and POST elements to generate the sentence. Defining for example a rule that matches a query for red objects as a HEAD element with the text "\textit{Objects of colour \{colour\}}" and a PRE rule with the text "\textit{\{colour\}}" enables the system to respond both with \textit{Objects of colour red} and \textit{Red objects}. [5] [4] As this paper focuses more on the evaluation than on the actual generation, a more detailed explanation of the NLG component of C-Phrase is omitted and the reader is instead referred to reading the papers by Michael Minock mentioned in the bibliography.
Part II

METHOD
CONFIGURATION

4.1 SETTING UP THE DATABASE

To get a reasonable big database fourteen different images were chosen. The images were chosen with three particular aspects in mind, they should only contain clear relations eg. behind over under, they should contain easily named colors, avoiding patterns, they should contain clear actors and object ie. the background should be as plain as possible without distractions that one might interpret as an object. These three rules were set up make the migration of the information in the images to the database as easy as possible, also to minimize misinterpretations of the pictures and the contents of these. This resulted in pictures that had at least one clear actor, human or animal, as well as a number of objects such as clothes or furniture.

The database schema, present in the appendix in C-Phrase compatible syntax, describe the objects and actors as well as the relations between the actors and objects.

Examples of this is the relation ACTOR_ON_OBJECT, referring to one actor and one object that said actor is on top of or sitting on. The database used was PostgreSQL, since it is ODBC-compatible and well tested with C-Phrase.

After creating the database and the schema the tables were filled manually in the way that the authors interpreted the pictures. This was also done in C-Phrase syntax directly into text files that the system reads and then creates and populates the tables.

4.2 CONFIGURING C-PHRASE

The next step was the configuring of C-Phrase, which means naming relations, attributes and joins, enabling the NLP quite quickly to understand basic queries such as “Actors”, “Scenes” and “Objects”. C-Phrase then understand what the query means and can start asking the database these simple queries, but it still can’t generate any answer strategy.

After this initial naming, C-Phrase listed more specific queries in it’s onthology tab and it became possible to do some more interesting building of semantic grammars. In the onthology, “PRE”, “HEAD”,
“POST” and “ANSWER” rules were configured giving C-Phrase answer strategies and an expanded grammar, although not extensively so.
EVALUATION METHOD

5.1 PROBLEMS WITH EVALUATION

There are several issues making evaluations difficult, for some issues you can evaluate a system by automated means. This, provided that you have a timely system for the evaluation, gives a lot of benefits in the form of reproducibility and efficiency. The goal of this thesis however is to evaluate human understanding of computer generated natural language query descriptions. An automated system to test human understanding is bound to rely heavily on some psycholinguistic model. If the model is bad, the results will be bad.

A survey with human respondents has other issues, some people can, wanting well, to help the authors of the paper and try to answer “correctly”, for instance by saying that they prefer a text that is obviously computer generated over a text that is written by hand. Comparing different sets of text can also be misleading because of differences in taste. Most of the respondents to this paper might be of Nordic and Swedish descent and what is regarded as good English is likely to differ between Stockholm and Oxford.

Another issue of evaluation is that if it is too technical, respondents might be limited to a small subset of the population that are relatively homogeneous. In the context of this paper such a population could be people who know SQL or people who are familiar with Codd’s Tuple Calculus.

5.2 OUR SURVEY METHOD

The survey method that this paper’s results will be based on tries to circumvent these problems. Trying to find out how well people understand generated query descriptions necessitates the use of real respondents to avoid using a potentially flawed model. In order to get good results enough respondents to be able to make statistically valid claims is required. There should be enough respondents so that they can be split into a control group that answer handwritten query descriptions and a group that answers generated descriptions. Whether a person was assigned to the generated text group or the human written group was randomized.
5.2 OUR SURVEY METHOD

A survey on the internet makes it easier to get many answers, especially if you have no funding to pay respondents, which we sadly did not. Internet surveys however come with a few issues, people are probably less likely to follow a link and answer an evaluation online then they are when asked in person to answer an evaluation. Even when potential respondents have clicked the link there is a risk that they will leave upon finding the survey too complex or suspecting it will take too much of their time. Efforts to reduce these effects were made by keeping the evaluation fairly short with only ten questions, informing the respondents directly that the survey would take only five minutes and keeping a progress bar at the top so people would have a clear sight of the end of the survey.

In order to alleviate the issue of a too technical survey, any notions of presenting respondents with SQL or tuple calculus was dropped. Instead they were presented with pictures representing scenes in the database. The respondent was told to choose between them and select the ones that they thought best matched the provided query description. All descriptions matched a query such as \( \{ x \mid \text{people}(x) \land x.hat = "red" \} \), and the text describing this might look like the pictures of people that have a hat of red colour. The questions and queries of the survey can be found in the appendix, as well as the schema used to describe the contents of the pictures.

Afterwards the survey respondent were presented a button to share the evaluation on social networks in order to get more respondents. There were thoughts of also presenting the user with a score, showing how well they did to match the queries, but it was decided against implementing this out of fear that it might skew the results by respondents redoing the survey to earn a higher score. See Figure 1 for a screenshot of one of the questions in the survey.

![Figure 1.: The interface for the survey.](image)
5.3 Survey Subjects

The survey was distributed on the internet for anyone to partake and shared on Facebook and other social sites. The survey was completely anonymous and no questions were asked about the respondents age group, gender or nationality. This because the audience for such a system is a varied and wide group of people, dividing the respondents into subgroups based on such indicators has no value when this paper wants to draw conclusions regarding the group as a whole, disregarding gender age etc. As the survey had both a group answering the NLG questions and a control group and the respondents had no control over which group they were put in, the groups are still comparable between each other and the results still of interest.

5.4 Survey Calculations

Both the time and answers for each of the 10 questions were recorded. When calculating the score for one question, both the correctly selected and the correctly unselected pictures yielded points, a score of maximum 14 points could be obtained on each question.
Part III

RESULTS
RESULTS

There were 113 people who participated in the survey. 62 of these filled out the survey with the human written text describing the queries, the rest, 51 people, filled out the survey with the generated texts. The results of the score of both the control group and the NLG group can be seen in Figure 2.

In question 1 the generated query outperformed the human written query and in question 2 both groups performed equal. On the rest of the questions the human written queries outperformed the generated ones, but only very marginally.

The control group and the NLG group do not differ more than 10 percent e.g. 65% for the human written one on question 5 and 56% for the generated one. The only question that this does not apply to is the 4th question.

In that particular case the results for the generated query was considerably worse than the human written query, 90% against 60% to be precise making this question stand out as the difference between groups in other queries are within a 10 percent span.

Furthermore the results of the time measurements are displayed in figure 3. Noticeable is that the performance across the two sets are varied. Although the human written one is generally the one with shortest answer time there are three cases where the generated one had shortest answer time.

Apart from that the answer times varied from question to question, it is noticeable that even when the human written queries had a longer answer time than the generated queries, the score was still in favour of the human written query.

The individual, more detailed results of question 1, 4 and 9 are given their own graphs since these questions are unique in either that the results differ from the overall tendencies or that the answer time was of interest.

Figure 4 shows the detailed results of question 4, the percentage number here representing the ratio of respondents choosing picture x and total respondents answering the question. Question 4 had a significant difference between the score of the control group and the
Results

Figure 2.: Displaying the average percentual correctness of respondents for all 10 questions. Set 0 is the human written queries and Set 1 the generated queries.

Figure 3.: Displaying the time respondents took to answer for each of the 10 questions. Set 0 is the human written queries and Set 1 the generated queries.

NLG group as seen in Figure 2.
The control group did much better here, and we would like to suggest to the reader to compare question 4 in the appendix of the different sets. It is noticeable that, except for picture 5 and 6, 7% of the people in the control group seem to have chosen all the pictures whereas for the NLG group, this number is around 35%.

Furthermore, the control group had 93% select the correct image, number 5, whereas the NLG group only had 75%. This will be discussed further under discussion.

![Figure 4](image)

Figure 4.: Displaying how many percent of the respondents on question 4 selected a certain picture as matching. Pictures are on the X-axis, Set 0 is the human written queries and Set 1 the generated ones.

The results of question 1 was the only question where the respondents of the generated text had the best score by a margin of 9%, see Figure 5. Seven of the correct answers had a margin of ≈10% by the respondents of the generated text set.

The incorrect pictures were chosen > 10% of the time by both groups apart from picture 14 that was chosen ≈12% by the respondents of the human written text set. The other pictures are within a ≈5% difference between the sets.

Question 9 was the question with the lowest correctness percentage and the longest answer time, see Figure 6. The answer frequency by picture of this question is varying wildly, the difference in correct answer selection are disparate from picture 4, where all correct answers are within 1% from each other, in the favour of the human written set.

Although this indicates that set 0 should be the set that is most cor-
Figure 5: Displaying how many percent of respondents selected which pictures on question 1. Set 0 is the human written queries and Set 1 the generated ones.

rect there are a lot of pictures chosen that are incorrect that counter this, for example pictures 1, 2 and three, which are incorrect answers, but still chosen at a much higher rate with the human written text than with the generated one.

Figure 6: Displaying how many percent of respondents selected which pictures on question 9. Set 0 is the human written queries and Set 1 the generated ones.
DISCUSSION

7.1 POSSIBLE IMPROVEMENTS

The survey could be improved by giving the respondents a score afterwards, a form of gamification of the survey. To avoid problems of people redoing the survey and skewing the results, the survey could use a cookie to track if the respondent has already answered or optionally, require that the respondent enter their email.

To share the survey, respondents had to enter a captcha when sharing on Facebook. For a future evaluation, research on how to avoid captchas should be done. Probably it is possible by having Facebook staff screen your app.

Perhaps also a more well-defined metric for understandability could create more portable and easily digested results. A possible metric could be \(\frac{\text{Score}}{\text{Time in seconds}}\), although care has to be taken that this metric actually is meaningful.

Regarding the data set and choice of images, it would be better to choose much less detailed pictures, rather cartoons or photos taken in a sterile empty environment. There were cases of us missing that there was a small white object somewhere in the picture, or another actor in the background. Not all people notice this and hence, respondent give different responses.

Another problem with the data set and yet again, the images, was the clarity. Some images had objects which some deemed were of beige colour rather than brown, Fclothes were not of one single colour but had patterns. If a chance had occurred to redo the evaluation, it might had been an option to even just describe geometric shapes.

Given more time, it would be interesting to do the evaluation twice, once with the basic, minimum configuration of C-Phrase and one where a set amount of extra man hours had been applied into making the texts even better.

It would also be interesting to measure more than just the understanding of the queries, which text the users prefer and think is more natural is also an interesting metric. This might partially be derived from the time it takes respondents to answer, if it takes very long they either had a cup of coffee, the text was difficult to understand or the question was difficult to understand. Obviously, it is not as accurate
7.2 WHAT WENT WELL

Despite there being room for improvement in the survey itself, the results of the survey gave a quite clear result. When a respondent is abstracted from having to actually know SQL to respond and when the evaluation is a step further from boring, the amount of people that will answer it and become interested in the subject increases.

The process of the survey, where the users have to choose the data the queries are supposed to describe as opposed to choosing the text that best describes the query, is a property that removes quite a lot of uncertainty. The latter process is subject to all kinds of human quirks, such as choosing the obviously computer generated text as the best text, only to please the authors. The process this paper devised is not devoid of such human quirks, but there are certainly less of them.

7.3 CONCLUSIONS

Regarding the survey results, in Figure 4 the hand written query description was “Pictures of both humans and animals” and the generated description “Pictures of the “human” and “animal” actors”. The respondents of the generated text were considerably worse. This likely stems from the natural interpretation of “and” which is more close to the logical “or”.

In the handwritten text the pattern “both X and Y” generate a natural interpretation that is almost equivalent to logical “and”. There is no difficulty whatsoever to have C-Phrase add the word “both”. Interestingly this error was introduced by the authors, thinking in technical and logical terms and interpreting a natural “and” as a logical “and”.

There can be a few things that can be learned from this. It might for instance be better to have a non-technical person author the system to avoid such misinterpretations. That choice of course depend also upon the target group. Another point this brings forward is the need for the administrator of C-Phrase to get feedback of user queries.

One way of doing this is to do employ a similar evaluation as done in this paper before the launch of the system into the real world. Another, bringing more long term usefulness to the administrator would be a tab in the C-Phrase admin system that lists failed queries or queries that did not return what a user expected.
7.3 CONCLUSIONS

The authors did not spend an extreme amount of resources to create the configuration of C-Phrase. The part that took longest was probably the actual population of the database which might not be a problem for an organization already sitting on a gigantic database.

The amount of actual configuration of C-Phrase needed in order to create results that are so comparable in terms of understandability to a manually written text shows that C-Phrase is a very capable system that can be implemented without gigantic costs.

Several respondents showed quite an interest in this thesis, wondering how well they did and what the thesis was about. It created much more enthusiasm than standard generic surveys. With some improvements to the survey it wouldn’t be a bad idea to run it on future NLIs to databases that are similar to C-Phrase.

The survey method could be useful before launching a system as a test. It would be interesting to see how a modification of the survey could be applied to different data sets that can be harder to express as images.

As noted, this paper only measures the understandability of generated queries in a minimal configuration of C-Phrase. How natural the language seemed was not measured, but despite that, C-Phrase shows capabilities beyond the authors expectations.

With a slightly longer configuration it would not be unfathomable to see C-Phrase services in the wild, at the least as a research project. The C-Phrase paper mentions having laid the foundation for plugging in a machine learning algorithm to increase accuracy after the initial authoring, which would be extremely interesting.

There are some parts of C-Phrase needing attention before going into the wild. A simplified manual that still brings up all the topics of authoring, fixing some stability issues regarding the web interface and improving upon the interface design, i.e. make the link to naming of joins more verbose.

This paper concludes that although the generated text of a minimal configuration of C-Phrase generally was outperformed by manually written texts, the margin it was outperformed by was much smaller than anticipated and that the generated texts were almost as understandable as texts written by humans.
SCHEMA

(SCENE (ID NAME) :STRING (NAME) :INT (ID) :KEY (ID))

(ARCHITECTURE (ID SCENE TYPE) :STRING (TYPE) :INT (ID) :KEY (ID) :REFERENCES ((SCENE SCENE)))

(OBJECT (ID SCENE CLASS TYPE COLOUR) :STRING (CLASS TYPE COLOUR) :INT (ID SCENE) :KEY (ID) :REFERENCES ((SCENE SCENE)))

(ACTOR (ID SCENE TYPE GENDER AGE HAIR_COLOUR) :STRING (TYPE GENDER HAIR_COLOUR) :INT (ID AGE SCENE) :KEY (ID) :REFERENCES ((SCENE SCENE)))

(ACTOR_ON_OBJECT (ID ACTOR OBJECT) :INT (ID ACTOR OBJECT) :KEY (ID) :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_UNDER_OBJECT (ID ACTOR OBJECT) :INT (ID ACTOR OBJECT) :KEY (ID) :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_BESIDE_OBJECT (ID ACTOR OBJECT) :INT (ID ACTOR OBJECT) :KEY (ID) :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_IN_FRONT_OF_OBJECT (ID ACTOR OBJECT) :INT (ID ACTOR OBJECT) :KEY (ID) :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_BEHIND_OBJECT (ID ACTOR OBJECT) :INT (ID ACTOR OBJECT) :KEY (ID) :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_WEARING_OBJECT (ID ACTOR OBJECT) :INT (ID ACTOR OBJECT) :KEY (ID) :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))
(ACTOR_USING_OBJECT (ID ACTOR OBJECT)
 :INT (ID ACTOR OBJECT) :KEY(ID)
 :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_EATING_OBJECT (ID ACTOR OBJECT)
 :INT (ID ACTOR OBJECT) :KEY(ID)
 :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_HOLDING_OBJECT (ID ACTOR OBJECT)
 :INT (ID ACTOR OBJECT) :KEY(ID)
 :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))

(ACTOR_INSIDE_OBJECT (ID ACTOR OBJECT)
 :INT (ID ACTOR OBJECT) :KEY(ID)
 :REFERENCES ((ACTOR ACTOR) (OBJECT OBJECT)))
1. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, OBJECT AS Y1
   WHERE
     x.id = y1.scene AND
     y1.class = 'furniture'

2. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, OBJECT AS Y1
   WHERE
     x.id = y1.scene AND
     y1.colour = 'brown'

3. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, ACTOR AS Y1,
     ACTOR_USING_OBJECT AS Y2,
     OBJECT AS Y3,
     ACTOR_ON_OBJECT AS Y4,
     OBJECT AS Y5
   WHERE
     x.id = y1.scene AND
     y1.gender = 'female' AND
     y1.age < 18 AND
     y1.id = y2.actor AND
     y2.object = y3.id AND
     y3.type = 'trumpet' AND
     y1.id = y4.actor AND
     y4.object = y5.id AND
     y5.type = 'sofa' AND
     y5.colour = 'white'

4. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, ACTOR AS Y1, ACTOR AS Y2
   WHERE
     x.id = y1.scene AND
     y1.type = 'human' AND
5. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, OBJECT AS Y1
   WHERE
     x.id = y1.scene AND
     y1.type <> 'human'

6. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, ACTOR AS Y1,
     ACTOR_ON_OBJECT AS Y2,
     OBJECT AS Y3, ACTOR AS Y4,
     ACTOR_UNDER_OBJECT AS Y5,
     OBJECT AS Y6
   WHERE
     x.id = y1.scene AND
     y1.gender = 'female' AND
     y1.id = y2.actor AND
     y2.object = y3.id AND
     y3.class = 'furniture' AND
     x.id = y4.scene AND
     y4.type = 'dog' AND
     y4.id = y5.actor AND
     y5.object = y6.id AND
     y6.type = 'table'

7. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, ACTOR AS Y1, OBJECT AS Y2
   WHERE
     x.id = y1.scene AND
     y1.gender = 'male' AND
     x.id = y2.scene AND
     y2.colour = 'brown'

8. SELECT DISTINCT X.ID, X.NAME
   FROM SCENE AS X, ACTOR AS Y1,
     ACTOR_INSIDE_OBJECT AS Y2,
     OBJECT AS Y3
   WHERE
     x.id = y1.scene AND
     y1.type = 'cat' AND
     y1.id = y2.actor AND
     y2.object = y3.id AND
9. SELECT DISTINCT X.ID, X.NAME
FROM SCENE AS X
WHERE
  x.id NOT IN (  
    SELECT DISTINCT Y1.SCENE
    FROM ACTOR AS Y1
    EXCEPT(  
      SELECT DISTINCT Y2.SCENE
      FROM ACTOR AS Y2,
      ACTOR_WEARING_OBJECT AS Y3,
      OBJECT AS Y4
      WHERE
        y3.actor = y2.id AND
        y3.object = y4.id AND
        y4.type = 'pants' AND
        y4.colour= 'black'
    )  
  )

10. SELECT DISTINCT X.ID, X.NAME
    FROM SCENE AS X, OBJECT AS Y1,
     OBJECT AS Y4, ACTOR_ON_OBJECT AS Y3,
     ACTOR AS Y2
    WHERE
      x.id = y1.scene AND
      y1.type = y4.type AND
      y1.colour = 'brown' AND
      y2.type = 'dog' AND
      y3.object = y4.id AND
      y3.actor = y2.id AND
      y4.colour = 'red'
HANDWRITTEN QUESTIONS

1. Pictures of furniture
2. Pictures containing brown objects
3. Pictures of little girls using a trumpet on a white sofa
4. Pictures of both humans and animals
5. Pictures containing objects of white colour
6. Pictures of females on furniture and the pictures of dogs under tables
7. Pictures of both males and brown objects
8. Pictures of cats inside containers
9. Pictures without actors not wearing black pants
10. Pictures of brown objects that are of the same type as the red object a dog is sitting on
1. Pictures of the “furniture” objects
2. Pictures of the objects of colour “brown”
3. Pictures of the young “female” actors using object of type “trumpet” on a “white” “sofa” object
4. Pictures of the “human” and “animal” actors
5. Pictures of the objects of colour “white”
6. Pictures of the “female” actors on the “furniture” objects and pictures of “dog” actors under “table” object
7. Pictures of the “male” actors of the objects of colour “brown”
8. Pictures of the “cat” animal actors inside the “container” objects
9. Pictures not of actors not wearing “black” pants objects
10. Pictures of the objects of colour “brown” of type as object of colour “red” under “dog” actors
BIBLIOGRAPHY


