tion

16th World Computer Congress 2000

Proceedings of Conference on

Software: Theory and Practice

August 21 – 25, 2000 Beijing, China

Edited by
Yulin Feng
David Notkin
Marie-Claude Gaudel

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
WWW.PHEI.COM.CN

EXECUTIVE COMMITTE

General Chairs:

Kurki-Suonio Reino (*Tampere University, Finland*)

Jafu Xu (*Nanjing University, China*)

Program Chairs:

Yulin Feng (*Institute of Software/ Academia Sinica, China*)

David Notkin (*University of Washington, USA*)

Marie-Claude Gaudel (*Univ. de Paris-Sud, France*)

Secretary:

Jun Wei (*Institute of Software/ Academia Sinica, China*)

PROGRAM COMMITTEE

Daniel Berry (*Univ. of Waterloo, Canada*)

Robert Bowdidge (*Intrinsa Corp.of Mountain View, USA*)

Christian Breiteneder (*Univ. of Vienna, Austria*)

Ed Brinksma (*Univ. of Twente, Netherlands*)

Alan Burns (*University of York, UK*)

Marsha Chechik (*Univ. of Toronto, Canada*)

To-yat Cheung (*Hong Kong City University*)

Piotr Dembinski (*Polish Academy of Sciences, Poland*)

Michel Diaz (*LAAS-CNRS, France*)

Jose Luis Fiadeiro (*Univ. of Lisbon, Portugal*)

John Fitzgerald (*Univ. of Newcastle, UK*)

Carlo Ghezzi (*TU Milan, Italy*)

Michael Gorlick (*Aerospace, USA*)

Andrzej M. Goscinski (*Deakin University, Australia*)

Armando Haeberer (*Pontificia Univ. Catolica do Rio de Janeiro, Brazil*)

Xudong He (*Florida Intel. Univ.*)

Connie Heitmeyer (*US Navy Research, USA*)

Matthias Jarke (*TU Aachen, Germany*)

Cliff Jones (*Univ. of Newcastle, UK*)

Mathai Joseph (*TRDDC, India*)

Jan van Katwijk (*TU Delft, Netherlands*)

Kouichi Kishida (*SEA, Japan*)

Kai Koskimies (TU Tampere, Finland)

Yue-Sun Kuo (*Academia Sinica, Taiwan*)

Kevin Lano (*Imperial College, UK*)

Mingshu Li (*Chinese Academy of Sciences, China*)

Jian Lu (*Nanjing University, China*)

Robyn Lutz (*Iowa State U., USA*)

Joergen Steensgaard Madsen (*TU Denmark, Denmark*)

Boris Magnusson (*Lund University, Sweden*)

Paulo C. Masiero (*Univ. Sao Paulo, Brazil*)

Hong Mei (*Peking University, China*)

Peter Mosses (*University of Aarhus, Denmark*)

Jim Ning (*Andersen Consulting, USA*)

Fernando Orejas (*Universitat Politecnica de Catalunya, Spain*)

Jaan Penjam (*Institute of Cybernetics, Estonia*)

Klaus Pohl (*University of Essen, Germany*)

Igor Pottosin (*A.P. Ershov Institute, Russia*)

Michael Schrefl (*Univ. of South Australia, Australia*)

Harvey Siy (*Bell Labs, USA*)

Paul Sorenson (*U. Alberta, USA*)

Peter Szeredi (*IQSOFT Intelligent Software, Hungary*)

Tetsuo Tamai (*University of Tokyo, Japan*)

A Min Tjoa (*TU Vienna, Austria*)

Michael VanHilst (*HP Labs, USA*)

Shan Wang (*Chinese People University, China*)

Herbert Weber (*Fraunhofer Institute, Germany*)

Chaochen Zhou (*UNU/IIST, Macao*)

16th World Computer Congress 2000

Proceedings of Conference on Software: Theory and Practice

Hosted by

Chinese Institute of Electronics

China Computer Federation

China Institute of Communications

In Cooperation With:

IEEE Computer Society

图书在版编目（CIP）数据

2000年第16届世界计算机大会论文集．软件理论与应用卷 = Proceedings of Conference on Software: Theory and Practice/电子学会编．－北京：电子工业出版社，2000.8

ISBN 7－5053－6110－4

Ⅰ.2... Ⅱ.电... Ⅲ.①电子计算机－国际学术会议－文集－2000②软件－国际学术会议－文集－2000

Ⅳ.TP3

中国版本图书馆CIP数据核字(2000)第66899号

Acknowledgements

Main Sponsors of

the International Conference on Software: Theory and Practice (ICS2000)

the 16th World Computer Congress

August 21－25, Beijing China

IFIP TC2

International Institute of Software Technology / United Nations University

Institute of Software / Chinese Academy of Sciences

Publishers: **Publishing House of Electronics Industry**

Address: Wanshou Road, Beijing, China

Box: 173, Beijing, China, 100036

International Federation for Information Processing

ISBN 7－5053－6110－4/TP・3255

ISBN 3－901882－04－9

Printed in Beijing, China

Massive Information Systems on the Internet

Posters

MEASURES FOR ASSESSING SQL CODE[1] MAINTAINABILITY

MARIO PIATTINI[+] , ANTONIO MARTÍNEZ[+§]
[+]RESEARCH GROUP ALARCOS
DEPARTAMENTO DE INFORMÁTICA
UNIVERSITY OF CASTILLA-LA MANCHA
RONDA DE CALATRAVA, 5
13071, CIUDAD REAL. SPAIN.
Fax:00-34-926-221703 E-MAIL: {mpiattin, amartinez} @inf-cr.uclm.es

[§]EXCMA. DIPUTACION DE CIUDAD REAL
CALLE TOLEDO, 17
13001, CIUDAD REAL (SPAIN)

Abstract

Nowadays, relational databases are introduced in most of the Information Systems, becoming their essential core. The relational database language, SQL, is being increasingly used for application development.

Software engineers have been putting forward huge quantities of measures for software products, processes and resources. Unfortunately, almost all the measures proposed until now, have focused on 3GL program characteristics disregarding databases and their associated languages. Only a few author has proposed some metrics for 4GL effort estimation.

In this article we describe three simple measures for assessing SQL code maintainability. Both an experiment with students and a real case in a state owned organization have empirically validated these measures as maintainability indicators.

Keywords

Software measurement, maintainability, SQL, empirical software engineering.

[1] This research is part of the MANTICA project, partially supported by CICYT and the European Union (1FD97-0168).

. Introduction

Jowadays, relational databases are introduced in most f the Information Systems, becoming their essential ore. The relational database language. SQL[2], is used a almost all the organizations which have anagement information systems, either embedded in GL (Third Generation Language), mainly COBOL, rograms or in more modern productivity-enhancing ools such as 4GL (Fourth Generation Languages).

nformation Technology (IT) organizations must mprove the quality of software products, especially aintainability, because maintenance cost are the ost important problem of software development, anging between 60 and 90 percent of life-cycle costs Card and Glass, 1990; Pigoski, 1997). Software easurement is widely recognized as an effective eans to understand, monitor, control, predict and mprove software development and maintenance rojects (Briand et al., 1996). Software engineers have een putting forward hundreds quantities of measures r software products, processes and resources Fenton and Pfleeger, 1997; Melton, 1996). nfortunately, almost all the measures proposed until ow, have focused on 3GL program characteristics isregarding databases and their associated languages Sneed and Foshag, 1998), or object oriented nvironment (Henderson-Sellers,1996). Only some orks attempt to estimate the effort of developing 4GL rograms (Bourque and Côte, 1991; Dolado, 1997; erner and Tate, 1988) but we have not came accross ny efforts to try to assess SQL code maintainability. aintainability is achieved by three factors: nderstandability, modifiability and testability (Li and hen, 1987).

In this paper, three simple measures for assessing QL program maintainability are proposed. In next ction we describe the three measures. In section 3, n empirical validation experiment is presented. In ction 4 the results of a real case study are exposed. inally, in section 5 we summarize the paper and resent the conclusions.

. Proposed measures for assessing sql ode

QL language is composed of different kinds of atements: definition statements (e.g. CREATE ABLE), manipulation statements (e.g. INSERT) and ontrol statements (e.g. COMMIT). Four different anipulation statement can be found: SELECT, NSERT, DELETE and UPDATE. The first one is the ost used and so we focus our work on it.

e propose the following three measures for haracterizing SELECT statement:

T measure

Number of tables referred in the SELECT statement.

NN measure

Number of nesting in the SELECT statement. A SELECT statement without nested selects inside is NN=1.

NG measure

Number of GROUP BY in the SELECT statement.

These measures were proposed based on intuition and experience with SQL programs development and maintenance. The number of tables is likely to influence all the three maintainability factors as SQL statements will be more difficult to understand, to modify and to test if they include more tables. The number of nesting is likely to also influence the maintainability of the SQL code, as each nesting demands a new level of thinking similar to a new call in 3GL or a level of inheritance in object-oriented programs (Cant et al., 1995). The earlier relational optimizers had also been influenced by the SELECT nesting, and vendors recommended not to nest beyond 3 levels for performance reasons.

We believe that grouping rows for calculating values also influences maintainability of SQL programs as it implies an additional operation which must be carried out over a set of rows.

3. Empirical validation of the proposed measures.

In this section we summarize an experiment done in order to validate NT, NN and NG measures. This empirical validation has been carried out following the experimental method applied to software engineering (Bourque and Cote, 1991; Pfleeger, 1995).

Our aim is to demonstrate that the proposed measures can be used for measuring the understandability of the SQL statement which influences its maintainability.

3.1 Hypotheses

The formal hypotheses are:

- Null hypothesis: Different values of the three measures do not affect the understandability of the SELECT statement.
- Alternative hypothesis 1: The value of the NT measure affects the understandability of the SELECT statement.
- Alternative hypothesis 2: The value of the NN measure affects the understandability of the SELECT statement.
- Alternative hypothesis 3: The value of the NG measure affects the understandability of the SELECT statement.
- Alternative hypothesis 4: The combination of NT and NN measures affects the understandability of the SELECT statement.

[2] This paper focus on the version know as SQL'89.

```
select f.name_emp, p.number_fic, p.date
from control_employee p, employee f, h_employee h
where p.id_emp not in
      (select h.id_emp
     from control_employee p, employee f, h_employee h
       where p.number_fic=h.number_fic
             and f.id_emp=h.id_emp
             and p.id_emp=f.id_emp
             and p.date='171298'
             and p.control='SM'
             and p.status='A'
             and f.sex='V'
             and p.hour in
                 (select hour
     from control_employee p, employee f, h_employee h
                  where p.number_fic=h.number_fic
                        and f.id_emp=h.id_emp
                        and p.id_emp=f.id_emp
                        and p.date='171298'
                        and p.control='SM'
                        and p.tipe='A0'
                        and h.remaindert=0
                 )
      )
      and p.date='151298'
      and p.number_fic=h.number_fic
      and p.id_emp=f.id_emp
      and f.id_emp=h.id_emp
      group by f.name_emp, p.number_fic, p.date
```

Figure 1. Example of a SELECT statement (NT=3,NN=3,NG=1)

- Alternative hypothesis 5: The combination of NT and NG measures affects the understandability of the SELECT statement.
- Alternative hypothesis 6: The combination of NN and NG measures affects the understandability of the SELECT statement.
- Alternative hypothesis 7: The combination of NT, NN and NG measures affects the understandability of the SELECT statement.

3.2 Subjects

The participants of the field test were Computer Science students at the University of Castilla-La Mancha (Spain), who were enrolled in a database course lasting two semesters. Until the day of the experiment, the students did not known that they were to do it. The experiment was developed by 34 students, but only 19 of them were selected, because they answered rightly all the questions.

We have tried to minimize variability among participants by choosing people of the same degree, in particular from the third year.

3.3 Experimental materials

Eight separate SQL statements were required the hypotheses. In each one values of the measures were different. There were two p values for NT (one or three), two for NN (three) and two for NG (zero or one) documentation accompanying each design approximately twelve pages long including the and the queries. In appendix A the tables and th statements are shown.

The same table was used to reduce se variability in all the eight cases. In order tc learning errors, the eight cases were in differer for each subject, and the subjects were fo follow the order in which cases appeared experimental material.

The subjects were asked to write down the ini the final time, and the result for each query. V counted the time employed in giving correct an

3.4 Experimental design

Each level of one factor appears with each leve other one, so we have selected the crossing de x NN x NG. See table 1.

		FACTOR NT			
		LOW		HIGH	
		FACTOR NN			
		LOW	HIGH	LOW	HIGH
NG	LOW	1,1,0	1,3,0	3,1,0	3,3,0
	HIGH	1,1,1	1,3,1	3,1,1	3,3,1

Table 1. Crossed Design for the experiment

ıe the power of the test, α has been set to 0.1
0.05 level which is more common (Briand
7).

ırimental results

e type of experiment used, F statistic was
obtain the results.

7.5 software was used for the
ıs.Table 2 shows the results for the F-

ɔaring these values with $F_{1, 151} = 2.71$, we
: that:

ative Hypothesis 1: The value of the NT
res affects understandability of SELECT
ent. As 657.380>2.71, NT affects results of
ment, so that the alternative hypothesis 1 is

ative Hypothesis 2: The value of the NN
res affects understandability of SELECT
ent. As 456.514>2.71, NN affects results of
experiment, so that the alternative hypothesis 2 is valid.

- Alternative Hypothesis 3: The value of the NG measures affects understandability of SELECT statement. As 34.741>2.71, NG affect to results of experiment, so that the alternative hypothesis 3 is valid.
- Alternative Hypothesis 4: Combination of NT and NN measures affects understandability of SELECT statement. As 34.741>2.71, the interaction of NT and NN affects results of experiment, so that, the alternative hypothesis 4 is valid.
- Alternative Hypothesis 5: Combination of NT and NG affects understandability of SELECT statement. As 0.543<2.71, there is no significant effect of interaction between NT and NG.

ʻce of variation	Sum of Squares	DF	Mean Square	F	Sig of F
ʻce of ation	1.392,132	3	464,044	382,878	0
NT	796,737	1	796,737	657,380	0
NN	553,289	1	553,289	456,514	0
NG	42,105	1	42,105	34,741	0
ıy Interactions	43,184	3	14,395	11,877	0
NN	42,105	1	42,105	34,741	0
NG	0,658	1	0,658	0,543	0,462
NG	0,421	1	0,421	0,347	0,557
ıy Interactions	2,132	1	2,132	1,759	0,187
NN NG	2,132	1	2,132	1,759	0,187
ained	1,434,447	7	205,350	169,432	0
sual	174,526	144	1,212		
	1.611,974	151	10,675		

Table 2. Results of the experiment

- Alternative Hypothesis 6: Combination of NN and NG affects understandability of SELECT statement. As 0.421<2.71, there is no significant effect of interaction between NN and NG
- Alternative Hypothesis 7: Combination of NT, NN and NG affects understandability of SELECT statement. As 1.759<2.71, there is no significant effect of interaction between NT, NN and NG.

We can conclude that the three kinds of measures proposed have proved to be solid indicators of SQL programs understandability. These three measures are very easy to calculate and could be very useful to predict SQL maintainability.

4. Case study

4.1 General characteristics of the system.

The system is composed of 143 programs developed during a period of one year at the Data Processing Center of the state owned organization (Diputacion of Ciudad Real). The system is a transaction processing system for data maintenance. The programs with embedded SQL are all of small to medium size: each SELECT statement included an average of three tables, two-level nesting and one grouping.

One of the most positive aspects of the the fact that it was constructed complete same team, employing the same methodolo same developing environment (CA-OpenIn

4.2 Data analysis and results

4.2.1 Descriptive statistics

The general descriptive statistics for each variables are shown in table 3.

4.2.2 Correlation analysis

For the test of correlation we use coefficient statistics and Spearman's non-correlation to identify potentially re between the variable time of maintenance in minutes) and the measures defined, as relationships that could exist between variables. The results are shown in tables 4

The statistics of both correlation sets strong significant relationships between th time of maintenance and the measure defin for measure NG. We can observe relationships between the specificatio measured NT, NN and the variable maintenance are significant.

Variable	Mean	Variance	S.E. Skew	Low	High	Std Dev	Skewnes s
NT	6,042	30,364	0,203	0	22	5,510	1,361
NN	1,923	3,438	0,203	0	6	1,854	0,712
NG	0,399	0,241	0,203	0	1	0,492	0,419
TIME	74,364	5.838,865	0,203	1	290	76,413	1,205

Valid observations - 143 Missing observations – 0

Tabla 3: Descriptive statistics for each measure

		NT	NN	NG	TIME
Pearson	NT	1,000	0,796	0,280	0,986
	NN	0,796	1,000	0,258	0,881
	NG	0,280	0,258	1,000	0,284
	TIME	0,986	0,881	0,284	1,000
Sig	NT		0,000	0,001	0,000
	NN	0		0,002	0,000
	NG	0,001	0,002		0,001
	TIME	0	0,000	0.001	
N	NT	143	143	143	143
	NN	143	143	143	143
	NG	143	143	143	143
	TIME	143	143	143	143

Table 4. Pearson's Correlation Coefficients

		NT	NN	NG	TIME
Spearman	NT	1,000	0,799	0,286	0,964
	NN	0,799	1,000	0,243	0,908
	NG	0,286	0,243	1,000	0,283
	TIME	0,964	0,908	0,283	1,000
Sig	NT		0,000	0,001	0,000
	NN	0,000		0,003	0,000
	NG	0,001	0,003		0,001
	TIME	0,000	0,000	0.001	
N	NT	143	143	143	143
	NN	143	143	143	143
	NG	143	143	143	143
	TIME	143	143	143	143

Table 5. Spearman's correlation coefficients.

T and NG are also highly correlated, which is cal because each nesting introduces a table, which ılly is different from the table of the previous ing level. We are conscious that maintenance time depend on several other different factors than the ECT characteristics. Some programs have, les SELECT, other statements like INSERT, ETE or UPDATE; and also different procedural visual statements. However, due to the type of the rams involved, we think that these results can be lid first attempt to characterize SQL programs.

e level of grouping (NG) is not correlated with ime of maintenance. We cannot find an answer his now. More case studies and experiments must onsidered in order to explain the influence of ping in SQL understandability, modifiability and bility.

Conclusions and future works

re research is needed into the aspects of software urement (Neil, 1994), both from theoretical and practical points of view (Glass, 1996). We think very interesting to dispose of measures for onal databases. We have proposed and validated types of measures for assessing SQL program tainability: NT,NN,NG. We are also developing similar measures for other clauses of the CT statement such as "HAVING", "WHERE" ROM".

his firts step of the research we have focused on e SQL constructs. More complex SQL-92 on and Simon, 1993) and SQL:1999 (Eisenberg imon, 1999) sentences must be research.

re experiments and case studies are needed to m these measures as valid indicators for SQL am maintainability. Verification of these metrics ne formal frameworks as Briand et al. (1997) or (1998) is being carried out.

These measures are not enough to evaluate the maintainability of programs developed with 4GL, so different measures must be proposed for other different "sublanguages" besides data manipulation one. Traditional metrics must be adapted or new metrics must be defined to assess procedural, visual, control, definition and transaction statements.

Acknowledgements

This research has been supported by the MANTICA proyect CICYT and the European union under grant 1FD97-0168 . We thank paper to Fernando Brito e Abreu and Miguel Goulão for their thoughtful comment which contributed to improve considerably this paper.

References

Bourque, P. and Côté, V. (1991). An experiment in software sizing with structured analysis measures. *Journal of Systems and Software 15*, 159-172.

Briand, L., Bunse, D., Daly, J. and Differding, C. (1997). An experimental comparison of the Maintainability of Object-Oriented and Structured Design Documents, *Proc. Int. Conference on Software Maintenance, Harold, M.J. Visaggio, G. (eds), Bari, 1-3 Oct*, 130-138.

Briand, L.C., Morasca, S. and Basili, V. (1996). Property-based software engineering measurement. *IEEE Transactions on Software Engineering*, 22(1), 68-85.

Cant, S. N., Jeffery, D.R., and Henderson-Sellers, B.,(1995). A concept model of cognitive complexity of element of the programming process. *Information and Software Technology*. 37(7),351-362.

Card, D.N. and Glass, R.L. (1990). *Measuring Software Design Quality*. Englewood Cliffs. USA.

Dolado, J.J. (1997). A Study of the Relationships among Albrecht and Mark II Function Points, Lines of Code 4GL and Effort. *J. Systems Software,* 37, 161-173.

Eisenberg, A. and Melton, J. (1999). SQL: 1999, formerly know as SQL3. *SIGMOD Record,* 28 ,(1), 131-138.

Fenton, N. (1994). Software Measurement: A Necessary Scientific Basis. *IEEE Transactions on Software Engineering*, 20 (3), 199-206.

Fenton, N. and Pfleeger, S. L. (1997). *Software Measures: A Rigorous Approach* 2nd edition. London, Chapman & Hall.

Henderson-Sellers, B. (1996). *Object-Oriented Measures - Measures of complexity.* Prentice-Hall, Upper Saddle River, New Jersey.

Li, H.F. and Cheng, W.K. (1987). An empirical study of software measures. *IEEE Trans. on Software Engineering,* 13 (6), 679-708.

MacDonell, G., Shepperd, J. and Sallis, J. (1997). Measures for Database Systems: An Empirical Study. *IEEE Software,* 99-107

Melton, A. (ed.) (1996). *Software Mea* London, International Thomson Computer Pre

Melton, J. and Simon, A. (1993). *Understa new SQL: A complete guide.* Morgan K Publishers.

Pigoski, T.M. (1997). *Practical Maintenance.* Wiley Computer Publishin York, USA.

Pfleeger, S. L. (1995) Experimental De Analysis in Software Engineering. *Annals of Engineering 1,* 219-253. JC. Baltzer AG, Publishers.

Sneed, H.M. and Foshag, O. (1998). Measurin Database Structures. *Proc of The European Measurement Conference FESMA 98,* Antwer 8, Coombes, Van Huysduynen and Peeters (e 211.

Verner J. and Tate G. (1988). Estimating Effort in Fourth-Generation Developmen *Trans. on Software Engineering.* 15-22.

Zuse, H. (1998). *A framework of measurement.* Walter De Gruyter. 1998

)IX A

ELD DESCRIPTION

:SCRIPTION: employee

```
employee
ingres
24/03/1999 14:08:38
user table
OPING1.2
```

formation:

me	Type	Length	Nulls	Defaults	Key Seq
	varchar	9	no	no	1
	varchar	40	no	no	
	varchar	30	no	no	
	varchar	40	no	no	
:e	varchar	5	no	no	
	varchar	2	no	no	
	varchar	3	no	no	
	varchar	9	yes	null	
	varchar	1	no	no	
	varchar	1	no	no	
	varchar	2	no	no	
	date		no	no	
ι	varchar	2	no	no	
age	varchar	3	no	no	
try	varchar	3	no	no	
ial_sec	varchar	14	no	no	
ial_sec1	varchar	6	yes	null	

table "employee" has 4 rows.

:RIPTION: h_employee

```
h_employee
ingres
20/11/1998 18:05:17
user table
OPING1.2
```

rmation:

e	Type	Length	Nulls	Defaults	Key Seq
ervice	varchar	3	no	no	
	varchar	2	no	no	
	varchar	9	no	no	
	varchar	4	no	no	
	float	8	no	no	
	float	8	no	no	
	float	8	no	no	
	float	8	no	no	
	float	8	no	no	
	float	8	no	no	
	integer	1	no	no	
	integer	1	no	no	
	integer	2	yes	null	

```
subtipe                          integer        2  yes    null
key_emp                          varchar       15  yes    null
date_certificate                 varchar        6  yes    null
situation                        varchar        1  yes    null
remainderf                       date              yes    null
```

The table "h_employee" has 3 rows.

TABLE DESCRIPTION: control_employee

```
Name:                 control_employee
Owner:                ingres
Created:              04/03/1999 13:46:31
Type:                 user table
Version:              OPING1.2

Column Information:

                                                                  Key
Column Name                      Type       Length Nulls Defaults Seq
id_emp                              varchar          9   no      no
number_fic                       varchar          4   no      no
date                             varchar          6   no      no
hour                             varchar          4   no      no
code_incidence                   varchar          2  yes    null
control                          varchar          2  yes    null
status                           varchar          1  yes    null
code_center                      varchar          2  yes    null
tipe                             varchar          2   no      no
```

The table "control_employee" has 72 rows.

SQL STATEMENTS

```
1.- select hour from control_employee where number_fic='0959' and date='18129

2.- select number_fic, date, count(hour) as w_number_fic from control_employe
    where number_fic='0800' group by number_fic, date

3.- select number_fic, date from control_employee where number_fic not
    in
    (select number_fic from control_employee where date='171298' and
    control='SM' and status='A' and hour in (select hour from control_employe
    where date='171298' and control='SM' and tipe='A0')) and
    date>'131298'

4.- select number_fic from control_employee where number_fic not in
    (select Number_fic from control_employee where date='171298' and
    control='SM' and status='A' and hour in (select hour
    from control_employee where date='171298' and control='SM' and tipe='A0')
    and date>'131298' group by number_fic

5.- select f.name_emp, h.hour, p.key_emp from employee f, control_employee h,
    h_employee p where f.id_emp=h.id_emp and h.number_fic=p.number_fic
    and h.date='171298'

6.- select f.name_emp, p.key_emp from employee f, control_employee h,
    h_employee p where f.id_emp=h.id_emp and h.number_fic=p.number_fic
    and h.date='171298' group by name_emp, key_emp
```

```
elect f.name_emp, p.number_fic, p.date, h.date_certificate from
ontrol_employee, employee f, h_employee h where p.id_emp not in (select
.id_emp from control_employee p, employee f, h_employee h where
.number_fic=h.number_fic and f.id_emp=h.id_emp and p.id_emp=f.id_emp and
.date='171298' and p.control='SM' and p.status='A' and f.sex='V' and
.hour in (select hour from control_employee p,employee f, h_employee h
here p.number_fic=h.number_fic and f.id_emp=h.id_emp and p.id_emp=f.id_emp
nd p.date='171298' and p.control='SM' and p.tipe='A0' and h.remaindert=0))
nd p.date='151298' and p.number_fic=h.number_fic
nd p.id_emp=f.id_emp and f.id_emp=h.id_emp

elect f.name_emp, p.number_fic, p.date from control_employee p, employee
, h_employee h where p.id_emp not in (select h.id_emp from
ontrol_employee p, employee f, h_employee h where
.number_fic=h.number_fic and f.id_emp=h.id_emp and p.id_emp=f.id_emp and
.date='171298' and p.control='SM' and p.status='A' and f.sex='V' and
.hour in (select hour from control_employee p, employee f, h_employee h
here p.number_fic=h.number_fic and f.id_emp=h.id_emp and p.id_emp=f.id_emp
nd p.date='171298' and p.control='SM' and p.tipe='A0' and
.remaindert=0))and p.date='151298' and p.number_fic=h.number_fic and
.id_emp=f.id_emp and f.id_emp=h.id_emp
roup by f.name_emp, p.number_fic, p.date
```

第 16 届世界计算机大会
16th IFIP World Computer Congress

Beijing, China *21 - 25 August, 2000*

URL: http://www.wcc2000.org

ICCT 2000:
Prof. Gong Ke
Tel: 86 10 6278 4884
Fax: 86 10 6277 0317
E-mail: icct@wcc2000.org

ICDA 2000
Mr. Jiang Shiping
Tel: 86 10 6436 0984
Fax: 86 10 6436 4487
Email: icda@wcc2000.org

ICEUT 2000
Prof. Xu Zhuoqun
Tel: 86 10 6275 1798
Fax: 86 10 6275 1792
E-mail: iceut@wcc2000.org

ICS 2000
Prof. Feng Yulin
Tel: 86 10 6263 5241
Fax: 86 10 6256 2533
E-mail: ics@wcc2000.org

ICSP 2000
Prof. Yuan Baozong
Tel: 86 10 6324 0616
Fax: 86 10 6324 0616
E-mail:icsp@wcc2000.org

IIP 2000
Prof. Shi Zhongzhi
Tel: 86 10 6256 5533-5688
Fax: 86 10 6256 7724
E-mail:iip@wcc2000.org

ITBM 2000
Prof. Gan Renchu
Tel: 86 10 6891 2845
Fax: 86 10 6891 2483
E-mail: itbm@wcc2000.org

SEC 2000
Prof. Qing Shihan
Tel: 86 10 6263 5150
Fax:86 10 6256 2533
E-mail: sec@wcc2000. org

For Organizational Matters:
Mr. He Chuan
F/20, Bei Ao Plaza
No. 2 Jia, East Huixin St.
Chaoyang District
Beijing 100029, China
Tel: 86 10 6492 4744
Fax: 86 10 6492 4787
E-mail: organize@wcc2000.org

For Program Matters:
Prof. Li Xiao Ming
Dept. of Computer Science and Technology, Peking University
Beijing 100871, China
Tel: 86 10 6275 6231
Fax: 86 10 6275 1792
E-mail program@wcc2000.org

Dear author(s),

It is my pleasure to inform you that your paper titled
Author: MARIO PIATTINI, ANTONIO MARTíNEZ
Title: Measures for Assessing SQL Code Mainability
Paper ID: ICS-79
has been accepted as Normal Paper by the ICS2000 Program Committee of World Computer Congress 2000 to be held in Beijing, China, August 21-25, 2000.

For preparing the camera-ready papers, a notice for the camera-ready paper submission is enclosed in this letter. Please remember that your camera-ready papers may not be more than 8 pages for Normal Paper, 4 pages for Short Paper, 1 page for Poster (included in the proceedings as Extended Abstract). The extra pages have to pay sub-charge (USD 50/page). Authors are required to register and pay the registration fee at the time that the papers are submitted for publication in the proceedings. The registration form is enclosed, and can also be downloaded from the web site http://wcc2000.pku.edu.cn.

We believe that your participation in the conference will certainly enhance the conference importance and success. We are looking forward to meeting you at the conference in Beijing.

Thank you!

Sincerely Yours,

Yulin Feng

Program Co-Chair of ICS2000