

PONENCIAS

X CONGRESO NACIONAL  
USUARIOS DE ORACLE

A Coruña  
27 - 29 de Octubre 1999

**Core**  
Círculo de Usuarios Oracle de España

Círculo de Usuarios Oracle de España

# Cuore

**X CONGRESO NACIONAL  
USUARIOS DE ORACLE**

**A Coruña**  
27, 28 y 29  
Octubre 1999

Con el Patrocinio Especial de



**Sun**  
microsystems  
We're the dot in.com™

## Propuesta de métricas para la mantenibilidad de bases de datos ORACLE

**Mario Piattini, Coral Calero, Macario Polo y Francisco Ruiz**  
(Universidad de Castilla - La mancha)

Ronda de Calatrava, 5  
13071, Ciudad Real  
Tel.: 926 29 53 00 Fax. :926 29 53 54



- Ponencia disponible en Power Point - 248 Kb.
- Ponencia disponible en Word - 141 Kb.

---

### **RESUMEN DE LA PONENCIA**

---

Las métricas han demostrado ser bastante útiles para mejorar la calidad de los productos software, especialmente por lo que respecta a su mantenimiento que, como es sabido, representa el principal problema del desarrollo del software, suponiendo entre el 60 y 90% de los costes del ciclo de vida. Las métricas sirven como un medio eficaz para entender, monitorizar, controlar, predecir y mejorar los proyectos de desarrollo y mantenimiento del software.

Los ingenieros del software han propuesto en las tres décadas pasadas grandes cantidades de métricas para productos, procesos y recursos. Sin embargo, prácticamente todas las métricas propuestas desde el famoso número ciclomático de McCabe hasta la fecha, se han centrado en los programas, "despreciando" las características de las bases de datos. Pensamos que resulta de gran interés proponer métricas para los aspectos de datos debido a la influencia que las bases de datos han ido adquiriendo en el proceso de desarrollo y mantenimiento del software.

La mantenibilidad de un producto software viene determinada por tres factores: comprensibilidad, modificabilidad y facilidad de prueba, que, a su vez, se encuentran influenciados por la complejidad. Hay que tener en cuenta que conseguir una sola medida para la complejidad es prácticamente imposible, por lo que hay que distinguir diferentes tipos de complejidad. En esta ponencia, nos centraremos en la complejidad de producto, proponiendo algunas métricas para bases de datos ORACLE. Estas métricas no sólo están siendo verificadas en diversos marcos teóricos, sino que también se están validando tanto experimentalmente como en diversos casos de estudio llevados a cabo en colaboración con distintas empresas y organismos.

# PROPUESTA DE MÉTRICAS PARA LA MANTENIBILIDAD DE BASES DE DATOS ORACLE<sup>1</sup>

**Coral Calero, Mario Piattini, Macario Polo, Francisco Ruiz**

**Grupo ALARCOS**

Departamento de Informática  
Universidad de Castilla-La Mancha

Ronda de Calatrava, 5

13071, Ciudad Real

Tel.: +34 926 29 53 00 (ext. 3715)

Fax.: +34 926 29 53 54

e-mail: [ccalero@inf-cr.uclm.es](mailto:ccalero@inf-cr.uclm.es)

## 1. INTRODUCCIÓN

En los últimos años ha habido una gran difusión de las bases de datos relacionales, de forma que podemos encontrarlas tanto en ordenadores personales como en grandes computadores. Las bases de datos, y especialmente las relacionales, han pasado a ser el corazón de los sistemas de información aunque su diseño sea una tarea ardua y larga (De Miguel y Piattini, 1993) debido a los sólidos fundamentos matemáticos en los que se basa la definición del modelo relacional (Codd, 1970).

Gran parte de las metodologías existentes (Batini et al., 1992; Hansen y Hansen, 1997; Connolly y Begg, 1998; De Miguel et al., 1999; McFadden et al., 1999) dividen el diseño de bases de datos relacionales en tres etapas: modelado conceptual, modelado lógico y modelado físico. Además, mediante la aplicación de una serie de reglas, es posible transformar el esquema conceptual en relacional garantizando la calidad de este último mediante la teoría de la normalización. Pero aunque esta teoría haya demostrado su utilidad, resulta conveniente poder evaluar la calidad de las bases de datos relacionales.

En esta línea, las métricas han demostrado ser bastante útiles para mejorar la calidad de los productos software, especialmente por lo que respecta a su mantenimiento que, como es sabido, representa el principal problema del desarrollo del software, suponiendo entre el 60 y 90% de los costes del ciclo de vida. Las métricas sirven como un medio eficaz para entender, monitorizar, controlar, predecir y mejorar los proyectos de desarrollo y mantenimiento del software (Briand et al., 1996) además de guiar a desarrolladores e investigadores en la toma de decisiones (Pfleeger, 1997) y de ayudar a los diseñadores a elegir entre varias alternativas. Así pues, es imprescindible diseñar y disponer de nuevas métricas con tal de poder garantizar la calidad de los productos software.

---

<sup>1</sup> Este trabajo forma parte del proyecto MANTICA, parcialmente financiado por la CICYT y la Unión Europea (1FD97-0168)

Los ingenieros del software han propuesto en las tres décadas pasadas grandes cantidades de métricas para productos, procesos y recursos. Sin embargo y lamentablemente, desde que McCabe propuso la complejidad ciclométrica (McCabe, 1976) hasta nuestros días la mayor parte de las métricas se han centrado en las características de los programas ignorando por completo las bases de datos (Sneed y Foshag, 1998) debido a lo cual hay una auténtica necesidad de métricas específicas ya que el tamaño y la naturaleza de los datos tienen una influencia directa sobre muchos aspectos de los sistemas de información (Mac Donnell et al., 1997).

La mantenibilidad de un producto software viene determinada por tres factores (ISO 9126): comprensibilidad, modificabilidad y facilidad de prueba, que, a su vez, se encuentran influenciados por la complejidad (Li y Cheng, 1987). Sin embargo, intentar diseñar una medida general para medir la complejidad de las bases de datos relacionales es como perseguir el “santo grial” (Fenton, 1994) por lo que hay que distinguir diferentes tipos de complejidad.

Henderson-Sellers (1996) divide la complejidad en representacional, computacional y psicológica. Esta última está compuesta por la complejidad del problema, por los factores cognitivos humanos y por la complejidad del producto. Este último tipo es en el que se centra nuestra investigación. Así pues, este artículo trabaja con la complejidad de producto, proponiendo algunas métricas para bases de datos Oracle.

Habitualmente, las métricas propuestas para las bases de datos relacionales suelen ser genéricas por lo que resulta necesario adaptarlas a las características específicas de los Sistemas Gestores de Bases de Datos si quiere probarse su funcionalidad (Churcher and Shepperd, 1995). En este artículo hemos elegido Oracle para aplicar nuestras métricas debido principalmente a la importancia que este SGBD tiene en el mundo real ya que ofrece una tecnología líder que ha evolucionado, de forma que en su última versión incluye características de bases de datos objeto-relacional. En nuestro caso, además, aplicamos las métricas para medir el Diccionario de Datos Oracle.

## **2. MÉTRICAS PARA BASES DE DATOS RELACIONALES**

Desde que el Dr. Codd propusiera su modelo relacional a finales de los sesenta (Codd, 1970), el campo de las bases de datos han evolucionado y una muy importante industria se ha generado entorno a los productos relacionales.

Date define un sistema gestor de base de datos relacional como un “sistema en el que como mínimo:

- ♦ Los datos son percibidos por el usuario como tablas (y sólo como tablas); y

- ◆ Los operadores disponibles generan nuevas tablas desde las antiguas. Estos operadores incluyen, como poco, los operadores SELECT, PROJECT, JOIN” (Date, 1995).

El único indicador utilizado para medir la calidad de una base de datos relacional ha sido la teoría de la normalización, a partir de la cual Gray et al. (1991) proponen un ratio de normalización.

En este artículo proponemos una serie de métricas que, dependiendo del nivel a que las apliquemos, pueden ser divididas en métricas orientadas a tabla y métricas orientadas a esquema. De hecho, dos de las métricas, NA y RD, pueden aplicarse a ambos niveles.

- Como métricas orientadas a tabla proponemos las siguientes:

**Número de Atributos (NA(A), number of attributes)**

NA(A) es el número de atributos de la tabla A.

**Grado de Referenciabilidad (RD(A), referential degree)**

RD(A) es el número de claves ajenas de la tabla A.

- Como métricas orientadas a esquema proponemos las siguientes:

**Número de Atributos (NA, number of attributes)**

NA es el número de atributos de todas las tablas del esquema.

**Grado de Referenciabilidad (RD, referential degree)**

RD es el número de claves ajenas que hay en el esquema.

**Longitud Máxima del Árbol Referencial (DRT, depth referential tree)**

DRT se define como la longitud del camino referencial más largo de todos los caminos referenciales del esquema relacional. Los ciclos sólo se cuentan una vez.

**Cohesión del esquema(COS, cohesion of the schema)**

COS se define como la suma del número de tablas (al cuadrado) de cada subgrafo relacional en la base de datos:

$$COS = \sum_{i=1}^{|US|} NTUS_i^2$$

|US| número de subgrafos no relacionados

NTUS<sub>i</sub> número de tablas en el subgrafo no relacionado “i”

La verificación formal de las métricas presentadas (Calero et al., 1999b) ha sido realizada siguiendo el marco formal propuesto por Zuse (1998). También se ha realizado una validación empírica con las métricas relacionadas con la integridad referencial (Calero et al. 1999a) concluyendo de la misma que el número de claves ajenas es un indicador sólido de la entendibilidad del esquema de una base de datos relacional así como que la longitud máxima de los caminos referenciales ayuda a modular los efectos de la métrica RD.

### 3. APLICACIÓN DE LAS MÉTRICAS A LA BASE DE DATOS ORACLE

En esta sección presentamos la adaptación a Oracle realizada con las métricas de ambos tipos aquí presentadas, aplicándolas al Diccionario de Datos de Oracle. Para cada métrica presentamos la siguiente información:

- Descripción de la métrica
- Código SQL asociado para la obtención del valor para la métrica
- Resultados obtenidos al aplicar el código SQL al diccionario de Oracle

#### 3.1. Métricas orientadas a tabla

- *Número de atributos.* NA(A) es el número de atributos de la tabla A.

El código SQL necesario para obtener el valor de esta métrica es:

<pre>SELECT ATC.TABLE_NAME, COUNT(ATC.COLUMN_NAME) FROM ALL_TAB_COLUMNS ATC, ALL_TABLES AT WHERE ATC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS' GROUP BY ATC.TABLE_NAME;</pre>	<p>Dónde:</p> <p>ALL_TAB_COLUMNS es una vista que lista las columnas de todas las tablas, vistas y clusters accesibles por el usuario</p> <p>ALL_TABLES es una vista que contiene las descripciones de las tablas relacionales accesibles por el usuario</p> <p>SYS es el propietario del diccionario de datos Oracle.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Los resultados obtenidos al aplicar el código anterior al Diccionario de Datos Oracle son:

TABLE_NAME	COUNT (ATC.COLUMN_NAME)	TABLE_NAME	COUNT (ATC.COLUMN_NAME)	TABLE_NAME	COUNT (ATC.COLUMN_NAME)
ACCESS\$	4	HSS_FDS_INST	4	PROFNAME\$	2
AQS_MESSAGE_TYPES	7	HSS_GRANTED_USERS	3	PROP\$	3
AQS_QUEUE_STATISTICS	3	HSS_INST_CAPS	6	PSTUBTBL	6
ARGUMENT\$	22	HSS_INST_DD	5	REFCON\$	5
ATEMPTAB\$	1	HSS_INST_INIT	5	REG_SNAP\$	8
ATTRCOL\$	3	HSS_PRIVILEGES	3	RESOURCE_COST\$	2
ATTRIBUTES	15	ICOL\$	14	RESOURCE_MAP	3
AUD\$	28	IDL_CHAR\$	6	RESULTS	12
AUDIT\$	4	IDL_SB4\$	6	RGCHILD\$	5
AUDIT_ACTIONS	2	IDL_UB1\$	6	RGROUP\$	10
BOOTSTRAP\$	3	IDL_UB2\$	6	SEG\$	20
CCOL\$	11	ID_GEN\$	1	SEQ\$	9
CDEF\$	21	INCEXP	6	SLOG\$	8
CLU\$	26	INCFIL	5	SNAP\$	30
COL\$	24	INCVID	1	SNAP_COLMAP\$	7
COLLECTION\$	16	IND\$	34	SNAP_REFOP\$	8
COLTYPE\$	9	INDPART\$	27	SNAP_REFTIME\$	6
COM\$	3	JOBS	19	SOURCE\$	3
CON\$	9	KOPM\$	3	STMT_AUDIT_OPTI ON_MAP	2
DBMS_ALERT_INFO	4	LAB\$	3	SYN\$	4
DBMS_LOCK_ALLOCATE D	3	LIBRARY\$	4	SYSAUTH\$	4
DEFROLE\$	2	LINK\$	9	SYSTEM_PRIVILEG E_MAP	2
DEPENDENCY\$	7	LOB\$	16	TAB\$	37
DIR\$	3	METHOD\$	10	TABLE_PRIVILEGE_ MAP	2
DUAL	1	MIGRATE\$	3	TABPART\$	25
DUC\$	7	MLOG\$	11	TRIGGERS	14
ERROR\$	6	MLOG_REFCOL\$	5	TRIGGERCOL\$	5
EXPACK\$	10	NOEXP\$	3	TRUSTED_LIST\$	2
FET\$	4	NTAB\$	5	TSS	32
FILES	14	OBJ\$	21	TSQ\$	8
HISTGRM\$	8	OBJAUTH\$	8	TYPE\$	13
HIST_HEAD\$	17	OBJPRIV\$	2	TYPED_VIEWS	9

HSS_ACCESS_GRANTEDES	3	OID\$	3	TYPE_MISC\$	3
HSS_BASE_CAPS	2	PARAMETER\$	14	UETS	7
HSS_BASE_DD	3	PARTCOL\$	5	UGROUP\$	6
HSS_CLASS_CAPS	6	PARTOBJ\$	23	UNDOS	22
HSS_CLASS_DD	5	PENDING_SESSION\$	8	USER\$	25
HSS_CLASS_INIT	5	PENDING_SUB_SESSION\$	9	USER_ASTATUS_M	2
		\$		AP	
HSS_CREATE_GRANTEDES	2	PENDING_TRANS\$	23	USER_HISTORY\$	3
HSS_EXTERNAL_OBJECTS	6	PROCEDURES	4	VIEWS	8
HSS_FDS_CLASS	3	PROFILES	4	_default_auditing_opti	1
				ons_	

Asimismo es posible obtener los valores para la media, el máximo, el mínimo y la desviación estándar aplicando el siguiente código SQL:

<pre>SELECT AVG(COUNT(ATC.COLUMN_NAME)), STDDEV(COUNT(ATC.COLUMN_NAME)), MAX(COUNT(ATC.COLUMN_NAME)), MIN(COUNT(ATC.COLUMN_NAME)) FROM ALL_TAB_COLUMNS ATC, ALL_TABLES AT WHERE ATC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS' GROUP BY ATC.TABLE_NAME;</pre>	<p>Dónde:  ALL_TAB_COLUMNS es una vista que lista las columnas de todas las tablas, vistas y clusters accesibles por el usuario  ALL_TABLES es una vista que contiene las descripciones de las tablas relacionales accesibles por el usuario  SYS es el propietario del diccionario de datos Oracle.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Los resultados obtenidos al aplicar el código al Diccionario de Oracle son:

<pre>AVG(COUNT(ATC.COLUMN_NAME)) = 8.6829268 STDDEV(COUNT(ATC.COLUMN_NAME)) = 8.035084 MAX(COUNT(ATC.COLUMN_NAME)) = 37 MIN(COUNT(ATC.COLUMN_NAME)) = 1</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------

□ **Grado de Referenciabilidad.** RD(A) es el número de claves ajenas en la tabla A.

El código SQL necesario para obtener el valor de RD(A) es:

<pre>SELECT AC.TABLE_NAME, COUNT(AC.CONSTRAINT_NAME) FROM ALL_CONSTRAINTS AC, ALL_TABLES AT WHERE AC.CONSTRAINT_TYPE = 'R' AND AC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS' GROUP BY AC.TABLE_NAME;</pre>	<p>Dónde:  ALL_CONSTRAINTS es una vista que listalas definiciones de restricciones de las tablas accesibles.  ALL_TABLES es una vista que contiene las descripciones de las tablas relacionales accesibles por el usuario  SYS es el propietario del Diccionario de Datos de la Base de datos.  R es el tipo de definición de restricción para la integridad referencial.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Los resultados que se obtienen al aplicar este código al Diccionario de Datos son:

TABLE_NAME	COUNT (AC.CONSTRAINT_NAME)	TABLE_NAME	COUNT (AC.CONSTRAINT_NAME)
HSS_ACCESS_GRANTEDES	3	HSS_EXTERNAL_OBJECTS	1
HSS_CLASS_CAPS	2	HSS_FDS_INST	1
HSS_CLASS_DD	2	HSS_GRANTED_USERS	1
HSS_CLASS_INIT	1	HSS_INST_CAPS	2
HSS_CREATE_GRANTEDES	2	HSS_INST_DD	2
		HSS_INST_INIT	1



### 3.2. Métricas orientadas a esquema

- ❑ **Número de Atributos.** NA es el número de atributos de todas las tablas del esquema.

El código SQL necesario para obtener NA es:

<pre>SELECT COUNT(ATC.COLUMN_NAME) FROM ALL_TAB_COLUMNS ATC, ALL_TABLES AT WHERE ATC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS';</pre>	<p>Dónde: ALL_TAB_COLUMNS es una vista que lista las columnas de todas las tablas, vistas y clusters accesibles por el usuario ALL_TABLES es una vista que contiene las descripciones de las tablas relacionales accesibles por el usuario SYS es el propietario del diccionario de datos Oracle.</p>
---------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Los resultados obtenidos al aplicar el código al Diccionario de Datos son:

```
COUNT(ATC.COLUMN_NAME) = 1068
```

- ❑ **Grado de Referenciabilidad .** RD se define como el número de claves ajenas en el esquema.

El código SQL necesario para obtener el valor para esta métrica es:

<pre>SELECT COUNT(*) FROM ALL_CONSTRAINTS AC, ALL_TABLES AT WHERE AC.CONSTRAINT_TYPE = 'R' AND AC.TABLE_NAME = AT.TABLE_NAME AND AT.OWNER = 'SYS';</pre>	<p>Dónde: ALL_CONSTRAINTS es una vista que listalas definiciones de restricciones de las tablas accesibles. ALL_TABLES es una vista que contiene las descripciones de las tablas relacionales accesibles por el usuario SYS es el propietario del Diccionario de Datos de la Base de datos. R es el tipo de definición de restricción para la integridad referencial.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Los resultados obtenidos al aplicar el código anterior al Diccionario de Datos son:

```
COUNT(*) = 18
```

- ❑ **Longitud Máxima del Árbol Referencial.** DRT se define como la longitud del camino referencial más largo de todos los caminos referenciales del esquema relacional. Los ciclos sólo se cuentan una vez.
- ❑ **Cohesión del Esquema.** COS se define como la suma del número de tablas (al cuadrado) de cada subgrafo relacional en la base de datos:

$$COS = \sum_{i=1}^{|US|} NTUS_i^2$$

|US| número de subgrafos no relacionados  
NTUS<sub>i</sub> número de tablas en el subgrafo no relacionado "i"

Para calcular los valores de estas métricas, debemos exportar los datos a una aplicación Visual Basic. Previamente resulta necesaria una transformación en los datos de la base de datos. El proceso de transformación es el siguiente:

- Realizamos una selección de los datos de la base de datos y generamos cuatro ficheros ".dat". En cada uno de estos cuatro ficheros almacenamos la información acerca de: tablas (**tables**), restricciones de las tablas (**table\_constraints**), columnas (**columns**) y columnas clave (**key\_column\_usage**). La tabla 1 muestra como se realiza la selección de los datos para obtener los mencionados ficheros.

<b>Tables.dat</b>	
<pre>SELECT AAT.TABLE_NAME, AAT.TABLE_TYPE FROM ALL_ALL_TABLES AAT, ALL_TABLES ALT WHERE AAT.TABLE_NAME = ALT.TABLE_NAME AND ALT.OWNER = 'SYS';</pre>	<p>Dónde :</p> <p>ALL_ALL_TABLES es una vista que describe todas las tablas (tablas objeto y relacionales) accesibles por el usuario.</p> <p>ALL_TABLES es una vista que contiene descripciones de tablas relacionales accesibles por el usuario.</p> <p>SYS es el propietario del diccionario de datos de la base de datos.</p>
<b>Tables_constraints.dat</b>	
<pre>SELECT AC.CONSTRAINT_NAME, AC.CONSTRAINT_TYPE, AC.TABLE_NAME FROM ALL_CONSTRAINTS AC, ALL_TABLES ALT WHERE AC.TABLE_NAME = ALT.TABLE_NAME AND (AC.CONSTRAINT_TYPE = 'P' OR AC.CONSTRAINT_TYPE = 'R') AND ALT.OWNER = 'SYS';</pre>	<p>Dónde :</p> <p>ALL_CONSTRAINTS es una vista que lista las definiciones de las tablas accesibles.</p> <p>ALL_TABLES es una vista que contiene descripciones de las tablas relacionales accesibles por el usuario.</p> <p>SYS es el propietario del diccionario de datos de la base de datos.</p>
<b>Columns.dat</b>	
<pre>SELECT ATC.TABLE_NAME, ATC.COLUMN_NAME, ATC.COLUMN_ID FROM ALL_TAB_COLUMNS ATC, ALL_TABLES ALT WHERE ATC.TABLE_NAME = ALT.TABLE_NAME AND ALT.OWNER = 'SYS';</pre>	<p>Dónde :</p> <p>ALL_TAB_COLUMNS Es una vista que lista las columnas de todas las tablas, vistas y clusters accesibles por el usuario.</p> <p>ALL_TABLES es una vista que contiene las descripciones de las tablas relacionales accesibles por el usuario.</p> <p>SYS es el propietario del diccionario de datos de la base de datos .</p>
<b>Key_column_usage.dat</b>	
<pre>SELECT ACC.CONSTRAINT_NAME, ACC.TABLE_NAME, ACC.COLUMN_NAME, ACC.POSITION FROM ALL_CONS_COLUMNS ACC, ALL_CONSTRAINTS AC, ALL_TABLES ALT WHERE ACC.TABLE_NAME = AC.TABLE_NAME AND AC.TABLE_NAME = ALT.TABLE_NAME AND ACC.CONSTRAINT_NAME = AC.CONSTRAINT_NAME AND (AC.CONSTRAINT_TYPE = 'P' OR AC.CONSTRAINT_TYPE = 'R') AND ALT.OWNER = 'SYS';</pre>	<p>Dónde :</p> <p>ALL_CONS_COLUMNS es una vista que contiene información sobre las columnas accesibles en las definiciones de restricciones.</p> <p>ALL_CONSTRAINTS es una vista que lista las definiciones de restricciones en tablas accesibles.</p> <p>ALL_TABLES es una vista que contiene descripciones de las tablas relacionales accesibles por el usuario.</p> <p>SYS es el propietario del diccionario de datos de la base de datos.</p>

Tabla 1. Selección de los datos

- Con los datos seleccionados (los cuatro ficheros “.dat”) la aplicación Visual Basic crea un esquema de cuatro tablas (figura 1) e incluye en cada una de ellas los datos obtenidos con anterioridad (en cada tabla se incluirá la información de cada uno de los cuatro ficheros “.dat”).

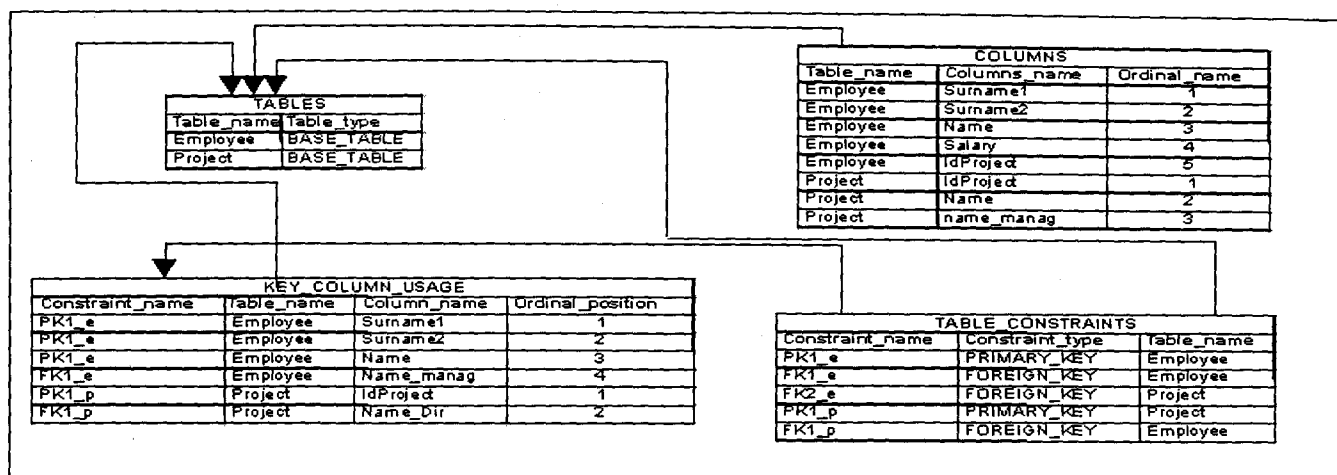


Figura 1. Esquema creado a partir de los ficheros generados

Con el esquema generado, la aplicación Visual Basic calcula los valores para las métricas COS y DRT utilizando la teoría de grafos.

Al aplicar el código a las tablas obtenidas del Diccionario de Datos Oracle, obtenemos los siguientes valores para las métricas:

COS = 235
DRT = 3

#### 4. CONCLUSIONES Y TRABAJO FUTURO

Creemos que es necesario abordar con mayor rigor los aspectos relativos a la calidad de las bases de datos relacionales para lo que resulta imprescindible utilizar una serie de métricas específicas que guíen al diseñador, ayudando a elegir entre diferentes esquemas alternativos. Es necesaria una mayor investigación en métricas de software (Neil, 1994) tanto desde el punto de vista teórico como práctico (Glass, 1996). Además es necesario adaptar las métricas a SGBD específicos.

En este artículo se ha propuesto un conjunto de métricas para medir la complejidad del esquema de bases de datos relacionales y poder de esta forma disponer de un mecanismo para probar y asegurar su calidad. Sin embargo no basta con proponer métricas, también es necesario someter las mismas a procesos de verificación formal y validación empírica para poder asegurar su utilidad. En esta línea, las métricas presentadas no sólo están siendo verificadas en diversos marcos teóricos, sino que también se están validando tanto

experimentalmente como en diversos casos de estudio llevados a cabo en colaboración con distintas empresas y organismos.

Igualmente resulta imprescindible aplicar las métricas a bases de datos específicas. En este artículo hemos utilizado código SQL para seleccionar los parámetros necesarios del Diccionario de Datos Oracle y poder de esta forma obtener los valores para nuestras métricas.

Este trabajo se complementa con otras métricas para bases de datos activas (Díaz y Piattini, 1999) y para bases de datos objeto-relacionales (Piattini et al. 1998).

Como parte del trabajo futuro, pretendemos adaptar nuestras métricas a otros SGBD que siguen el modelo SQL2. El objetivo final es poder ofrecer una guía precisa de utilización de las métricas así como desarrollar una herramienta para la recolección automática de las mismas.

Otro de los proyectos futuros de trabajo consistiría en la creación de sistemas de predicción para proyectos de bases de datos a partir de la utilización de estas y otras métricas.

## BIBLIOGRAFÍA

- ◆ Batini, C., Ceri, S. y Navathe, S., (1992), *Conceptual database design. An entity relationship approach*. Benjamin Cummings Publishing Company.
- ◆ Briand, L., Morasca, S. y Basili, V. "Property-Based Software Engineering Measurement". *IEEE Transactions on software Engineering*, Vol. 22, No 1, January, 1996.
- ◆ Calero, C., Piattini M., Polo, M y Ruiz, F. (1999a). Validating referential integrity as a database quality metric. *ICEIS 99*, pp 45-50, 27-31 March. Portugal.
- ◆ Calero, C. y Piattini, M. (1999b). Caracterización formal de métricas para bases de datos relacionales. *JISBD 99*, 24-26 Noviembre, Cáceres.
- ◆ Churcher, N.J. y Shepperd, M.J. (1995), Comments on "A Metrics Suite for Object-Oriented Design. *IEEE Trans. on Software Engineering* Vol. 21, N° 3, 1995, pp. 263-265.
- ◆ Codd, E.F. (1970). A Relational Model of Data for Larged Shared Data Banks. *CACM*, 13 (6), 377-387.
- ◆ Connolly, T., Begg, C. y Strachan, A., (1998), *Database systems*, 2ª Ed. Addison-Wesley
- ◆ Date, C.J. (1995). *An Introduction to Database Systems*. 6<sup>th</sup>. Addison-Wesley, Reading, Massachusetts.

- ◆ De Miguel, A. y Piattini, M., (1993), *Concepción y diseño de bases de datos relacionales*, Ed. Addison-Wesley
- ◆ De Miguel, A., Piattini, M. y Marcos, E. (1999), *Diseño de bases de datos relacionales*, Ed. RA-MA
- ◆ Díaz, O. y Piattini, M. (1999). Metrics for active databases maintainability. Accepted in CAISE'99. Heidelberg, June 16-18.
- ◆ Fenton, N. (1994). Software Measurement: A Necessary Scientific Basis. *IEEE Transactions on Software Engineering*, 20(3): 199-206.
- ◆ Glass, R. (1996). The Relationship Between Theory and Practice in Software Engineering. *IEEE Software*, November, 39 (11), 11-13.
- ◆ Gray R.H.M., Carey B.N., McGlynn N.A. y Pengelly A.D., (1991), Design metrics for database systems. *BT Technology J*, Vol 9, 4 Oct, 69-79
- ◆ Hansen, G.W. y Hansen J.V., (1997), *Diseño y administración de bases de datos*, 2ª Ed. Ed. Prentice-Hall
- ◆ Henderson-Sellers, B. (1996). *Object-oriented Metrics - Measures of complexity*. Prentice-Hall, Upper Saddle River, New Jersey.
- ◆ ISO, (1994). *Software Product Evaluation-Quality Characteristics and Guidelines for their Use*. ISO/IEC Standard 9126, Geneva.
- ◆ Li, H.F. y Chen, W.K. (1987). An empirical study of software metrics. *IEEE Trans. on Software Engineering*, 13 (6): 679-708.
- ◆ McCabe, T.J. (1976). A complexity measure. *IEEE Trans. Software Engineering* 2(5): 308-320.
- ◆ MacDonell, S.G., Shepperd, M.J. y Sallis, P.J. (1997). Metrics for Database Systems: An Empirical Study. Proc. Fourth International Software Metrics Symposium – Metrics'97, Albuquerque. IEEE Computer Society, pp. 99-107.
- ◆ McFadden, F.R., Hoffer, J.A. y Prescott, M.B., (1999), *Modern database management*, 5ª Ed. Addison-Wesley
- ◆ Neil, M. (1994) Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality. *Software Quality Journal* 3 (2), 65-78.
- ◆ Piattini, M., Calero, C., Polo, M. y Ruiz, F. (1998). Maintainability in Object-Relational Databases. . *Proc of The European Software Measurement Conference FESMA 98*, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.), 223-230.
- ◆ Pfleeger, S. L. (1997). Assessing Software Measurement. *IEEE Software*. March/April, pp. 25-26.

- 
- ◆ Sneed, H.M. y Foshag, O. (1998). Measuring Legacy Database Structures. Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.), 199-211.
  - ◆ Zuse, H. (1998). *A Framework of Software Measurement*. Berlin, Walter de Gruyter.