

## MTSQL – LENGUAJE DE CONSULTA PARA EL SISTEMA ADMINISTRADOR DE BASES DE DATOS METASYS

**A. Morales P.**

**O. Mascaró G.**

Universidad Católica de Valparaíso  
Av. Brasil 2950, Valparaíso, Chile

### R E S U M E N

El Sistema Administrador de Bases de Datos METASYS constituye una implementación muy cercana al modelo de entidades y relaciones propuesto por P. Chen en 1976, que ha sido adaptado para su aplicación en instalaciones de tamaño reducido. El modelo de entidades-relaciones, sin embargo, puede también ser observado como un conjunto de tablas componentes de un modelo relacional, para efectos de la aplicación de un lenguaje de consulta tipo SEQUEL. Aprovechando las características propias del modelo empleado por METASYS, se ha diseñado un lenguaje de consulta cuyo formato es más simple que aquel de SEQUEL pues solo debe declarar la tabla objetivo, junto al camino trazado para la consulta y los predicados asociados a la búsqueda. El presente trabajo describe las características de METASYS además de las principales propiedades y limitaciones del lenguaje de consulta MTQL.

## 1.-INTRODUCCION

En atención al tamaño predominante de las instalaciones computacionales existentes en nuestro medio, el Centro de Computación de la Universidad Católica de Valparaíso decidió en 1976 orientar una parte importante de sus esfuerzos de investigación al desarrollo de herramientas que hiciesen más eficaz el proceso de construcción de software en instalaciones reducidas.

Como producto de este trabajo, ha sido publicada una serie de artículos donde se describe un conjunto de metodologías que han sido desarrolladas a la luz de las características, actitudes y aptitudes que exhibe el medio mencionado [1, 2, 3]. Estas metodologías han sido asimismo divulgadas ampliamente a través de cursos y seminarios, demostrando la validez de las hipótesis de trabajo que las sustentan.

Paralelamente al desarrollo de metodologías de producción de software, se ha sostenido un permanente interés en la automatización de los mecanismos de almacenamiento de información, derivando obviamente hacia el área de bases de datos [4, 5]. Como producto final de estos proyectos, se ha construido un sistema administrador de bases de datos denominado METASYS [6], inspirado principalmente en el modelo entidades-relaciones propuesto por Chen en 1976 [7].

El sistema administrador de bases de datos METASYS ha servido de base a la implementación de varias aplicaciones que actualmente se encuentran en normal explotación, demostrando en la práctica tener no solo importantes atributos desde el punto de vista de las facilidades que otorga al programador en cuanto al manejo intuitivo del modelo, sino además positivas ventajas relativas a la administración global de los datos de la instalación.

Una vez alcanzados los objetivos asociados a la disponibilidad de una herramienta destinada a la administración de datos (METASYS), ha sido necesario abordar el desarrollo de un mecanismo de consulta que, orientado preferentemente al usuario final, permita interrogar fácilmente la base de datos. Ello ha dado lugar al lenguaje MTSQL, el cual no es otra cosa que una adaptación de los conceptos empleados por aquellos lenguajes que se sustentan en el álgebra relacional [8] a los requerimientos del modelo METASYS. En las siguientes secciones se describirá con algún detalle las características del modelo METASYS, con el propósito de discutir a continuación las propiedades del lenguaje de consulta MTSQL.

## 2.-EL MODELO METASYS

Intuitivamente, es fácil advertir la existencia de entidades (concretas o abstractas) durante el proceso de modelación del mundo real cuando se diseña una base de datos. Asimismo, entre las entidades

definidas se deriva, casi naturalmente, un conjunto de relaciones que establecen las asociaciones existentes en el mundo real. Si se toma el ejemplo de un conjunto de datos donde participan alumnos y asignaturas, es posible distinguir la entidad "alumno" y la entidad "asignatura" como elementos independientes dentro del modelo, pero que pueden encadenarse mediante la relación "inscripción", la cual llevará implícita la propiedad de que un alumno tenga varias asignaturas inscritas y, a la vez, que una asignatura tenga varios alumnos inscritos (Fig. 1).

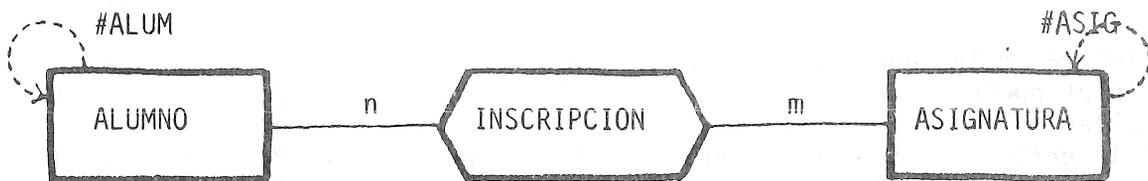


Figura 1.- Esquema básico entidad-relación

El sistema METASYS permite modelar relaciones m:n (BETWEEN-AND) como aquella del ejemplo considerado, así como relaciones del tipo 1:n (FROM-TO) que también se encuentran frecuentemente en la práctica (p. ej.: la relación asignatura-facultad, donde una facultad ofrece varias asignaturas, pero una asignatura dada solo es ofrecida por una sola facultad en forma exclusiva).

A cada relación es posible atribuir ciertas restricciones que sirven de base para otorgar una mayor consistencia al modelo con respecto del mundo real. En efecto, toda relación puede ser declarada como UNIQUE o NONUNIQUE, esto es, atribuirle a cada posible combinación entre dos elementos determinados de dos diferentes entidades, la propiedad de establecer relaciones con una sola ocurrencia permitida (UNIQUE), o la aceptación de varias ocurrencias (NONUNIQUE). En el ejemplo de alumnos y asignaturas anteriormente presentado, la relación "inscripción" es del tipo UNIQUE por cuanto un alumno solo está autorizado para tener una sola inscripción en una asignatura dada. Sin embargo, si el modelo en diseño está destinado a almacenar información histórica, esto es, datos de varios semestres, la relación "inscripción" deberá ser declarada como NONUNIQUE, puesto que un alumno podrá inscribirse más de una vez en una misma asignatura en semestres diferentes, debido a la reprobación eventual de la asignatura en algún semestre anterior.

METASYS permite también modelar la presencia de relaciones que no tengan ningún contenido. La relación "inscripción" tiene una serie de atributos propios, tales como; fecha de inscripción, número del documento de inscripción y resultado final obtenido en la asignatura. Sin embargo, una relación como "asignatura-ofrecida", entre las entidades "facultad" y "asignatura", constituye una relación 1:n sin contenido propio, reconocida en METASYS como NULL.

Asimismo, la simple estructura jerárquica (FROM), también es modelable en METASYS, lo cual completa convenientemente las posibilidades del sistema para la representación apropiada de las situaciones encontradas habitualmente en la práctica del diseño de bases de datos. Este es el caso del eventual registro de las características de las distintas "evaluaciones" realizadas para cada asignatura durante el semestre, que constituye una ocurrencia múltiple de registros dependientes de una entidad.

Adicionalmente, METASYS presenta la propiedad de aceptar la especificación de índices secundarios, tanto para las entidades como para las relaciones, otorgando al modelo la capacidad de establecer múltiples puntos de entrada que atribuirán una mayor eficiencia a ciertas consultas a la base de datos. En el ejemplo de alumnos y asignaturas, es posible definir un índice secundario sobre la entidad "alumno" en base al atributo CEDULA-DE-IDENTIDAD, suponiendo que toda persona lo tiene asignado. Este índice será declarado como NODUP, esto es, que no aceptará duplicaciones puesto que se trata de un atributo cuya repetición sería errónea. Asimismo, es posible definir un índice secundario sobre la relación "inscripción", índice que a diferencia del anterior aceptará necesariamente duplicaciones (DUP), puesto que más de una inscripción podrá haberse efectuado en una misma fecha.

Una de las características relevantes del modelo entidades-relaciones, es su contribución en materia de representación gráfica de una base de datos. El sistema METASYS ha tomado como base las proposiciones que en este contexto formula Chen [7], adaptándolas a las particularidades de su propio modelo. La figura 2. presenta un esquema asociado al ejemplo que ha servido para ilustrar las características de METASYS en este trabajo. Una paralelogramo identifica a las entidades, mientras un hexágono identifica a las relaciones. El tipo de relación (n:m ó 1:n) queda explícitamente declarado en las líneas que acompañan a la relación. Asimismo, la propiedad de única (UNIQUE), no única (NONUNIQUE) o nula (NULL), queda directamente indicada sobre cada relación, salvo en el caso de la relación jerárquica, la cual va unida a una flecha desde su entidad dueña.

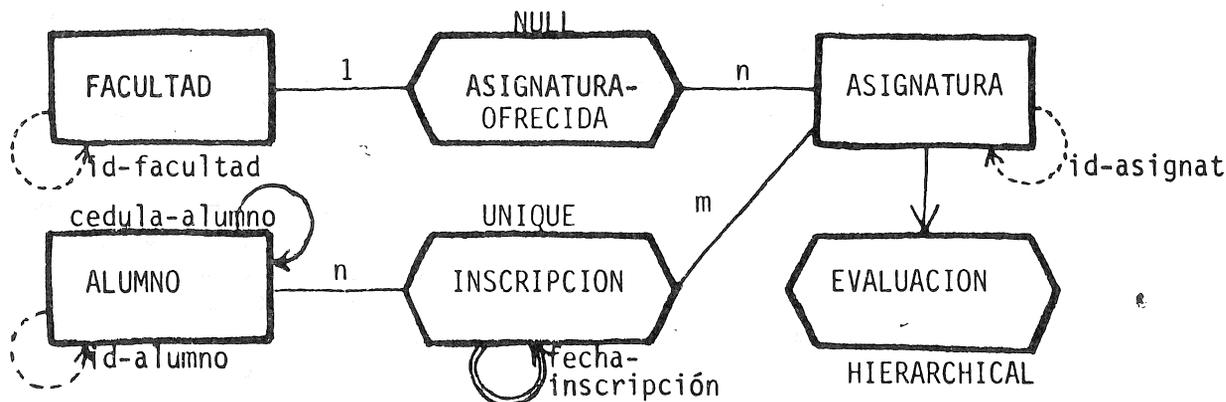


Fig. 2.- Representación gráfica del modelo.

La representación gráfica de los índices, se efectúa mediante semi-círculos dibujados sobre cada objeto indexado. La clave o índice primario, obligado para cada entidad, se declara mediante un semi-círculo con línea punteada. Un índice secundario que no acepta valores duplicados se representa mediante un semi-círculo en línea llena, y aquellas que aceptan valores duplicados, mediante una doble línea llena.

### 3.- LENGUAJES METASYS-DDL Y METASYS-DML

Toda base de datos modelada en función de las propiedades descritas para METASYS, puede ser expresada mediante el lenguaje de descripción de datos que el sistema dispone para ello (METASYS-DDL). Este lenguaje constituye una extensión a la estructura de la DATA DIVISION del lenguaje COBOL, en orden a proveer estructuras adicionales que reflejan el modelo METASYS. Para ilustrar con algún detalle las características de este lenguaje, se expresa a continuación la base de datos presentada en la Fig. 2.- :

DEFINE BASE: BASE-DE-DATOS-UNIVERSIDAD,BD&N.

ENTITY: FACULTAD, KEY=ID-FACULTAD.

```

01 REG-FACULTAD.
   03 ID-FACULTAD          PIC x(06).
   03 NOM-FACULTAD        PIC x(40).
      |
      |
      |
  
```

ENTITY: ALUMNO, KEY=ID-ALUMNO,  
INDEXED BY =(CEDULA-ALUMNO,NODUP).

```

01 REG-ALUMNO.
   03 ID-ALUMNO           PIC x(06).
   03 NOM-ALUMNO         PIC x(40).
   03 CEDULA-ALUMNO      PIC 9(10).
      |
      |
      |
  
```

ENTITY: ASIGNATURA, KEY=ID-ASIGNATURA.

```

01 REG-ASIGNATURA
   03 ID-ASIGNATURA      PIC x(06).
   03 NOM-ASIGNATURA     PIC x(40).
  
```

RELATION: ASIGNATURA-OFRECIDA, NULL  
FROM FACULTAD TO ASIGNATURA.

RELATION: INSCRIPCION, UNIQUE  
FROM FACULTAD TO ASIGNATURA.

RELATION: INSCRIPCION, UNIQUE  
BETWEEN ALUMNO AND ASIGNATURA  
INDEXED BY = (FECHA-INSCRIPCION, DUP).

01 REG-INSCRIPCION.

03 NUM-INSCRIPCION.

PIC 9(05).

03 FECHA-INSCRIPCION

PIC 9(06).

03 RESULT-INSCRIPCION

PIC x.

⋮

RELATION: EVALUACION, HIERARCHICAL  
FROM ASIGNATURA.

01 REG-EVALUACION.

03 TIPO-EVALUACION

PIC x.

⋮

Puede observarse que METASYS-DDL permite declarar la base (DEFINE BASE), así como cada entidad (ENTITY) y relación (RELATION) por separado. Toda entidad declarada exige la especificación de su clave primaria (KEY), aceptando adicionalmente la especificación de índices secundarios (INDEXED BY) que pueden ser del tipo duplicados (DUP) o del tipo que no aceptará duplicación de contenido (NODUP).

Las relaciones, a su vez, podrán ser declaradas en METASYS-DDL indicando su calidad (UNIQUE, NONUNIQUE, NULL, HIERARCHICAL) junto al tipo y definición de las entidades entre las cuales establece alguna asociación (FROM/TO para 1:n y BETWEEN/AND para n:m). El contenido de cada objeto así declarado, se expresa en términos normales del lenguaje, constituyendo METASYS-DDL una simple extensión de la DATA DIVISION. Esta especificación, sin embargo, no es incorporada directamente a ningún programa de aplicación, sino que pasa a formar parte del DICCIONARIO DE DATOS de la instalación, desde donde será posteriormente extraída durante el preprocesamiento de los programas de aplicación que contengan expresiones del lenguaje METASYS-DML.

Para la manipulación de la información almacenada, METASYS provee el lenguaje METASYS-DML que opera bajo la modalidad de registro-a-la-vez (record-at-a-time), inspirado principalmente en aquellos lenguajes desarrollados para el tratamiento de modelos de redes. Este lenguaje está formado por un conjunto de verbos que configuran una extensión de la PROCEDURE DIVISION del lenguaje COBOL, agregando la cláusula SELECTDB en la

ENVIRONMENT DIVISION para declarar la presencia de una base de datos dada en el programa.

Con el propósito de actualizar la base de datos, METASYS-DML ofrece los verbos:

STORE  
MODIFY y  
DELETE

que permiten, respectivamente: ingresar, modificar o eliminar una entidad o una relación. Cada una de estas operaciones es llevada a cabo comprobando su completa consistencia con el modelo definido. Es imposible, por ejemplo, eliminar una entidad que tiene relaciones activas dependientes, o modificar una relación alterando el contenido de un índice secundario que derive en una violación de su condición de no duplicado.

Además de los verbos de actualización, METASYS-DML provee un conjunto de expresiones destinadas a permitir la navegación a través de las estructuras del modelo definido. Estas expresiones constituyen variaciones del verbo FIND, básicamente sustentadas en el siguiente repertorio de sentencias:

FIND  
FIND FIRST  
FIND NEXT  
FIND LAST  
FIND PRIOR

y están destinadas a establecer accesos a los datos almacenados a través de los múltiples caminos que hayan sido definidos para un modelo en particular.

El programador de aplicaciones construirá su programa en lenguaje COBOL, incorporando los verbos de METASYS-DML al código de la PROCEDURE DIVISION. Una fase de pre-procesamiento del programa, que se sustenta en el DICCIONARIO DE DATOS de la instalación, convertirá toda expresión de METASYS-DML en una invocación (CALL) a alguna función apropiada de los módulos componentes del sistema administrador básico. Dado que el pre-procesador "conoce" detalladamente el modelo de datos definidos, será capaz de diagnosticar toda inconsistencia semántica de las expresiones METASYS-DML incorporadas al programa.

#### 4.- LENGUAJE DE CONSULTA MTSQL

La disponibilidad de un lenguaje de las características de METASYS-DML, ofrece un conjunto de facilidades al programador de a-

plicaciones en el manejo de las estructuras de la base de datos definida, con la ventaja adicional de una permanente protección de la consistencia del modelo. Sin embargo, presenta una seria limitante, por cuanto toda consulta a la base de datos deberá ser únicamente respondida en base a programas especialmente construídos para ello.

Esta limitante puede ser subsanada mediante la presencia de un lenguaje que sea capaz de expresar el objetivo de la consulta, sin especificar la forma de llevar a cabo el procedimiento (non-procedural). Para ello, ha sido desarrollado el lenguaje MTSQL que opera sobre modelos definidos en el sistema METASYS, bajo una modalidad de archivo-a-la-vez (set-at-a-time).

El lenguaje MTSQL se inspiró primitivamente en aquellos lenguajes concebidos para el manejo de modelos relacionales, específicamente SEQUEL [9], en atención a la propiedad que presenta cualquier base de datos de ser observable como compuesta por un conjunto de tablas relacionales.

En efecto, si cada entidad o relación del modelo METASYS se considera como una tabla relacional, explicitando la existencia en cada relación de los campos claves de las entidades que vincula, es posible aplicar a este nuevo modelo las funciones clásicas del álgebra relacional (SELECT, JOIN, PROJECT).

Sin embargo, en consideración a la complejidad que presenta la expresión en SEQUEL de consultas que requieran sucesivas aplicaciones de la función JOIN, se ha optado por una forma de lenguaje que, manteniendo cierta similitud con la expresión relacional, aproveche las particulares características del modelo METASYS, principalmente aquellas que se derivan del carácter binario de las relaciones. Asimismo la implementación de MTSQL ha excluído la función PROJECT en atención al costo en recursos de hardware y tiempo de proceso que conlleva. De manera que una consulta en MTSQL se limita a:

- expresar la tabla objetivo,
- expresar el camino (PATH), en términos de elementos del modelo, que lleva implícita la función JOIN, y
- expresar los predicados que establecen las restricciones para cada elemento incluído en el camino.

Es así como tomando el ejemplo de la base de datos de la Fig. 2.-, la consulta: "EXHIBIR TODOS LOS ALUMNOS QUE SE ENCUENTRAN INSCRITOS EN LA ASIGNATURA "CCI400", se expresa en MTSQL como sigue:

```
EXHIBIT : (ID-ALUMNO, NOM-ALUMNO)
PATH    : ASIGNATURA (ID-ASIGNATURA = "CCI400')
```

INSCRIPCION  
ALUMNO

Esta consulta será satisfecha a través de la pantalla (EXHIBIT) en la siguiente forma:

| ID-ALUM | NOM-ALUM               |
|---------|------------------------|
| 037543  | ABARCA MORALES ALBERTO |
| 087583  | ZAMORA BASCUÑAN PEDRO  |
| :       | :                      |
| :       | :                      |

Obviamente, para una consulta como la presentada no se aprecia una diferencia substancial, en cuanto a complejidad de expresión, respecto a SEQUEL:

```
SELECT ID-ALUMNO, NOM ALUMNO
FROM ALUMNO
WHERE ALUMNO.ID-ALUMNO=INSCRIPCION.ID-ALUMNO
AND ID-ASIGNATURA="CCI400"
```

Sin embargo, en una consulta de mayor complejidad es posible destacar la ventaja de MTSQL. Así por ejemplo, si se quiere obtener "LAS ALUMNAS MAYORES DE 21 AÑOS INSCRITAS EN ASIGNATURAS OFRECIDAS POR LA FACULTAD "LEYES", su especificación en MTSQL queda como sigue:

EXHIBIT (ID-ALUMNO, NOM,ALUMNO,EDAD-ALUMNO)

```
PATH FACULTAD (ID-FACULTAD=LEYES )
ASIGNATURA-OFRECIDA
ASIGNATURA
INSCRIPCION
ALUMNO (SEXO-ALUMNO="F" AND
EDAD-ALUMNO>"21")
```

en tanto en SEQUEL tendría la siguiente forma:

```
SELECT ID-ALUMNO, NOM-ALUMNO,EDAD-ALUMNO
FROM ALUMNO
WHERE SEXO-AL="F" AND EDAD-ALUMNO > 21
AND ID-ALUMNO IN
(SELECT ID-ALUMNO FROM INSCRIPCION WHERE
INSCRIPCION.ID-ASIGNATURA=ASIGNATURA-OFRECIDA.ID-ASIGNATURA
AND
ID-FACULTAD="LEYES")
```

MTSQL considera además, para aquellas consultas que implican restricciones sobre atributos no indizados o que no consideran restricciones, la posibilidad de definir el "punto de entrada", a través de un índice al camino de acceso, lo cual implica expresar de paso un ordenamiento. Así por ejemplo, la consulta: "LISTAR TODOS LOS ALUMNOS QUE VIVEN FUERA DE VALPARAISO", puede ser expresada:

```
LIST      : (ID-ALUMNO, CEDULA-ALUMNO, NOMBRE-ALUMNO, CIUDAD-ALUMNO)
PATH      : ALUMNO (CIUDAD-ALUMNO NOT="VALPARAISO")
KEY       : CEDULA-ALUMNO, ASCENDING
```

En este caso el recorrido de la entidad se realizará por el índice definido sobre CEDULA-ALUMNO, obteniéndose el listado ordenado por este atributo ascendentemente (ASCENDING), pudiendo definirse también la alternativa contraria (DESCENDING).

## 5.- CONCLUSIONES

La implementación de MTSQL, de acuerdo a lo descrito en el presente trabajo, constituye una versión simplificada de un lenguaje de consulta, percibiéndose que se han omitido opciones de especificación que hubiesen podido incorporarse sin gran esfuerzo al lenguaje. Este es el caso de la especificación de operaciones aritméticas sobre ciertos atributos asociados a la consulta, o la especificación de ordenamientos diferentes a aquellos en que la tabla objetivo es producida. Para justificar estas simplificaciones, es necesario indicar que a lo largo del proyecto se ha adoptado permanentemente como criterio de desarrollo, el dotar al sistema únicamente de aquellas opciones absolutamente necesarias al usuario medio de instalaciones reducidas, intentando compatibilizar las múltiples posibilidades del modelo con la carencia real de recursos computacionales del medio señalado. Asimismo, ha existido una tendencia predominante a comprobar en el corto plazo la factibilidad y operatividad del modelo, a costa de la exclusión de funciones tal vez muy interesantes.

De acuerdo a esta estrategia de desarrollo, la totalidad de los componentes de METASYS han superado satisfactoriamente la etapa de prueba, encontrándose en normal funcionamiento una serie de aplicaciones que han permitido demostrar no solo la adaptabilidad práctica del sistema a los problemas del usuario final, sino además su simplicidad de manejo para los propósitos de análisis, diseño y programación.

A su vez, MTSQL se encuentra en la fase de experimentación, con el objeto de verificar las hipótesis de trabajo adoptadas res-

pecto de la conducta del usuario frente al lenguaje y frente a sus productos de información. Hasta este instante, no se perciben actitudes del usuario sustancialmente diferentes a aquellas manifestadas respecto de otros lenguajes de consulta como SEQUEL. Sin embargo, las restantes fases del proyecto que se comienzan a abordar: incorporación de nuevas funciones y soporte de lenguaje natural, indudablemente ayudarán a despejar las incógnitas que aún se mantienen en torno a este aspecto de la relación hombre-máquina.

Finalmente, cabe destacar los beneficios subsidiarios que aporta un proyecto de esta naturaleza en la forma de un enriquecimiento de la tecnología local en el área de ingeniería de software, puesto que durante su transcurso ha sido necesario desarrollar y consolidar una serie de herramientas computacionales, así como un conjunto de metodologías, que son sistemáticamente divulgadas por la Universidad en el medio industrial.

#### REFERENCIAS

- [1] A. Morales, L. Barra  
System Development Techniques for Small  
and Medium Size Installations  
Proceeding of The 15th SIGCPR Conference  
ACM, New York, 1977
- [2] A. Morales, O. Mascaró  
Economía y Estandarización en la Producción de Software  
Proceedings PANEL'78  
Universidad Católica de Valparaíso, 1978
- [3] L. Barra, A. Morales  
Diagramas de Flujo para Programación Estructurada  
Proceedings PANEL'79  
Universidad Católica de Valparaíso, 1979
- [4] A. Morales  
Comments on Small Database Implementation  
5th International Conference on Very Large Data Bases  
Rio de Janeiro, 1979

- [5] A. Morales  
Implementación de Bases de Datos en Instalaciones Reducidas  
Proceedings PANEL '79  
Universidad Católica de Valparaíso, 1979.
- [6] A. Morales, O. Mascaró, R. Morales  
METASYS: Herramientas de Software para la Implementación  
de Bases de Datos en Instalaciones Reducidas  
Proceedings PANEL '80  
Universidad Simón Bolívar, Caracas, 1980.
- [7] P. Chen  
The Entity-Relationship Model - Toward a  
Unified View of Data  
ACM Transaction on Database System  
New York, 1976
- [8] E.F. Codd  
Relational Completeness of Data Base Sublanguages  
Database Systems  
Convant Computer Science Symposia Series  
Prentice Hall, 1972
- [9] M.M. Astrahan, D.D. Chamberlain  
Implementation of a Structured English  
Query Language  
R J 1464 (# 22484) IBM Research Lab.  
San José, CA, 1974.