



# ***The UML and Data Modeling***

A Rational Software Whitepaper

---

## **Table of Contents**

Introduction.....	1
The UML Data Modeling Profile .....	1
Database .....	1
Schema .....	2
Table.....	2
Key .....	3
Index.....	4
Relationship.....	5
Column.....	6
Datatype .....	6
Constraint .....	6
Summary .....	8

## ***Introduction***

---

The power of the Unified Modeling Language is not limited to object oriented software development. More and more, the UML is being applied to other areas of software development, such as data modeling, enhancing practitioners' ability to communicate their needs and assessments to the rest of the team.

Data analysts primarily gather data out of documented business requirements. The database itself traditionally has been described by notations called entity relationship diagrams, using graphic representation that is similar but not identical to that of the UML.

The UML can be used to describe the complete development of relational and object relational databases<sup>1</sup> from business requirements through the physical data model. However, modeling of the physical data model must express a detailed description of the database. This is done using Rational's Data Modeling Profile for the UML<sup>2</sup>.

## ***The UML Data Modeling Profile***

---

This white paper describes in detail the Data Modeling profile for the UML as implemented by Rational Rose® Data Modeler, including descriptions and examples for each concept including database, schema, table, key, index, relationship, column, constraint and trigger. It is written for those who have experience in traditional data modeling as well as some familiarity with the UML.

### ***Database***

The Database is the system for data storage and controlled access to stored data. It is the biggest element a data model supports.

The relational database is the standard database supported by the Data Modeling UML profile. An object relational database, an extension of the relational database, is also supported by the UML profile.

The stereotype <<Database>>, when used as a UML component, defines a database. As a component the database must have a name.

In the component view, a database is a target dependent element, since it references a database type.

A corresponding icon can represent the stereotype as shown in Figure 1.



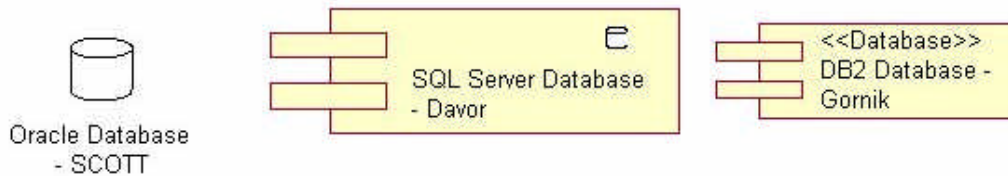
***Figure 1***

---

<sup>1</sup> Object oriented databases are supported by UML through modeling of persistent classes.

<sup>2</sup> Data Modeling Profile for UML is not yet approved by OMG.

The 3 possible representations of a Database component are shown in Figure 2.



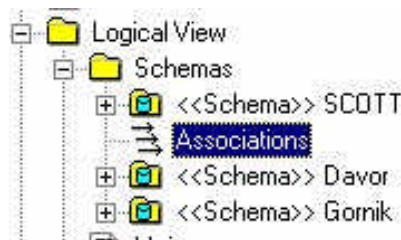
**Figure 2**

**Schema**

The full description of the data model to be used for retrieval and storage of data is stored in a schema inside of a database. The schema is the biggest unit that can be worked with at any given time .

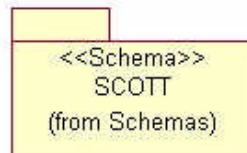
A package using the <<Schema>> stereotype in a UML model represents a database schema.

In a browser, a schema is shown as a package. (Figure 3)



**Figure 3.**

In a diagram, the representation is a package with the Schema stereotype. (Figure 4)



**Figure 4.**

Note that there can be more than one schema associated to a database.

**Table**

A table is the basic modeling structure of a relational database. It represents a set of records of the same structure, also called rows. Each of these records contains data. The information about the structure of a table is stored in the database itself.

A class with the <<Table>> stereotype represents a relational table in a schema of a database.

In the browser a table is represented by a table symbol. (Figure 5)

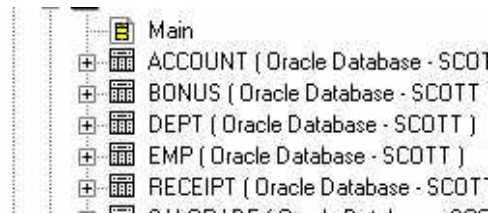


Figure 5

The representation in a diagram uses the Table stereotype and stereotype icon. (Figure 6)

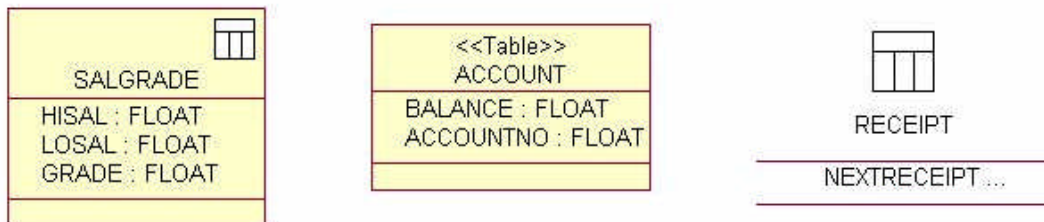


Figure 6

Hosting the table in the schema package creates the association of a table to a schema.

### Key

Keys are used to access the table. Primary keys uniquely identify a row in a table, while foreign keys access data in other related tables.

The key is represented as a key constraint and as a tagged value on the column.

A primary key uses a PK tag in front of the column as shown. (Figure 7)

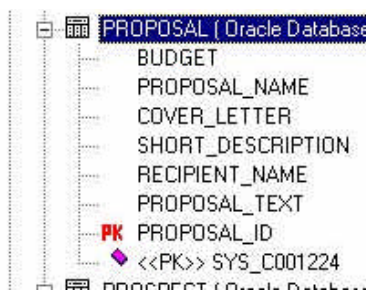
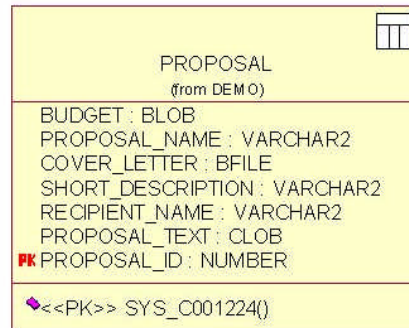


Figure 7

In a diagram, the PK tag represents the primary key, and the PK stereotyped operation is the primary key constraint. (Figure 8)



**Figure 8**

A foreign key represents a column, which is a part of a relationship to another table.

An FK tag represents the foreign key. It generates the foreign key constraint, which is represented by a stereotype FK on an operation.

Read about relationships for more on foreign keys.

**Index**

An Index is a physical data structure that speeds up data access. It does not change the quality or the quantity of data retrieved. An index can include multiple columns or just a single column.

The Index does not exist in the logical view.

The Index stereotype on an operation represents a key constraint for an index. (Figure 9)



**Figure 9**

The diagram representation uses the Index stereotype in front of the operation. (Figure 10)

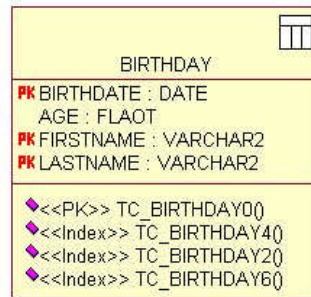


Figure 10

The index constraint specifies the columns included and optionally the uniqueness of the index.

**Relationship**

A dependency of any kind between tables in a data model is called a relationship.

A relationship is a summary of a stereotyped association and a set of primary and foreign keys.

Every relationship is between a parent and child table, where a parent table must have a primary key defined. The child table creates a foreign key column and foreign key constraint to address the parent table.

A non-identifying association represents a relationship between two independent tables. The foreign key of the child table does not contain all of the primary key columns. (Figure 11)

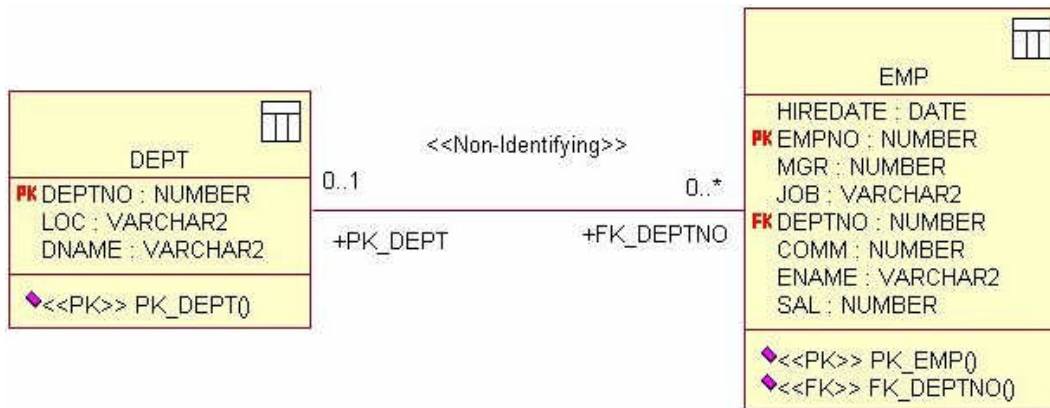


Figure 11

An identifying relationship is a relation between two dependent tables, where the child table cannot exist without the parent table. All of the primary keys of the parent table (Person, in this example) become both primary and foreign key columns in the child table (Account). (Figure 12)

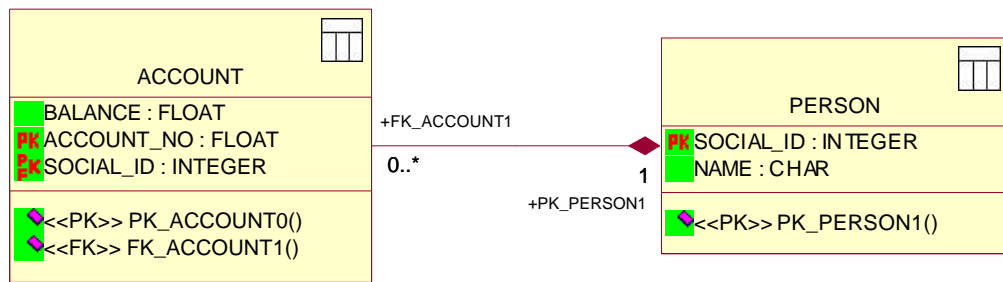


Figure 12

A relationship has two roles associated with it. They define the role of one table in association with the other. It is possible to assign more than one relationship between two tables using different roles.

Each relationship creates migrated keys from the parent to the child table.

**Column**

A table contains columns which are tagged attributes. Columns can contain data, when they are instantiated as a row.

A column must have a defined data type (see below for details).

The column can be either persistent or computed.

Computed columns are defined by an expression.

Persistent columns can be tagged as primary key columns, nullable columns, and unique columns. They can also contain a default value.

Constraints can be checked for any column.

**Datatype**

Supporting relational databases requires the support of standard datatypes.

Examples of datatypes are char, date, float, long, number, nvarchar2, raw, varchar2.

Object relational databases also require support of any enumeration designed in the model.

**Constraint**

A constraint is a rule applied to the structure of the database. This rule extends the structure and can be applied to a column and/or table.

All constraints are defined as stereotyped operations.

All the constraints described below are implemented here in one class. (Figure 13)



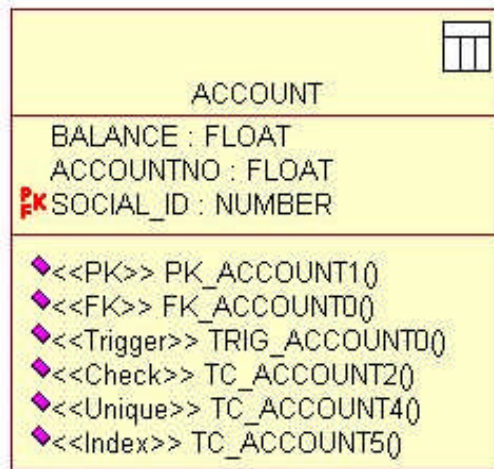


Figure 13

**Primary key**

The primary key constraint defines the primary key as a rule on the table.

**Foreign key**

The foreign key constraint defines the foreign key as a rule on the table.

**Trigger**

A trigger is an activity executed by the DBMS as a side effect or instead of a modification of a table or view to ensure consistent system behavior on data operations.

The Trigger stereotype on an operation represents the trigger on the table. (Figure 14)

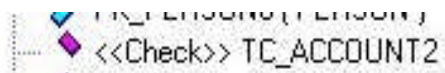


Figure 14

The trigger is displayed as a Trigger stereotype on the operation. (Figure 15)

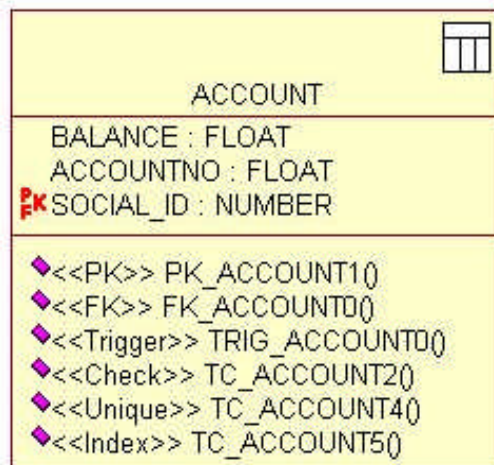


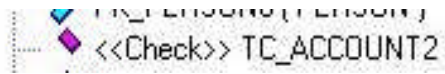
Figure 15

An example of a trigger is the logging of an insert into another table in the case where the balance is below 0 or greater than 100.000.

***Valid value***

The valid value constraint checks the value of data according to a given expression.

The Check stereotype on an operation represents the validation check constraint. (Figure 16)



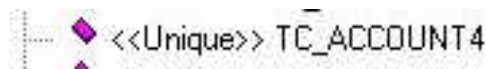
***Figure 16***

An example of the check constraint is BALANCE > 0.

***Uniqueness***

The unique constraint assures that every row contains a different value in a column.

The unique stereotype represents the uniqueness constraint. (Figure 17)



***Figure 17***

The unique constraint can be applied on composite columns and it can also exist on a single column.

***Summary***

---

With the data modeling for UML profile, the UML fully supports data modeling needs. It allows the support of software development and data modeling with one unified language. Using the UML data modeling profile, Rational Rose Data Modeler unifies software development teams with a single, shared tool.

# **Rational**<sup>®</sup>

**the e-development company**<sup>™</sup>

Corporate Headquarters

18880 Homestead Rd.

Cupertino, CA 95014

Toll-free: 800.728.1212

Tel: 408.863.9900

Fax: 408.863.4120

Email: [info@rational.com](mailto:info@rational.com)

Web site: [www.rational.com](http://www.rational.com)

For International Offices: [www.rational.com/corpinfo/worldwide/location.jtmpl](http://www.rational.com/corpinfo/worldwide/location.jtmpl)

The word Rational and Rational's products are trademarks of Rational Software Corporation. References to other companies and their products use trademarks owned by the respective companies and are for reference purposes only.

© Copyright 2000 by Rational Software Corporation.

TP-180 3/00 Subject to change without notice. All rights reserved