**LESSON 2 THE RELATIONAL DATA MODEL**

The Relational Database Model constitutes the theoretical basis of the commercial databases management systems more used nowadays.

The Relational Database Model is proposed by Ted Codd of IBM Research in 1970. The relational data model has its theoretical basis on the mathematical “Relationship Set Theory”, in which the data is logically structured in the form of relationships.

One of the main objectives of the Relational model is that the logical structure of data is independent of its internal storage, hence the model takes no account at all of the physical aspects of database, which is a responsibility of commercial DBMS manufacturers, whose guidelines are focused on the best performance of their systems.

The Relational Database Model may be considered as a combination of three components: structure, integrity and manipulation.

1. **Model Structure.**

The data structures are made up of tables or relations, rows or tuples, columns or attributes and primary keys and foreign keys.

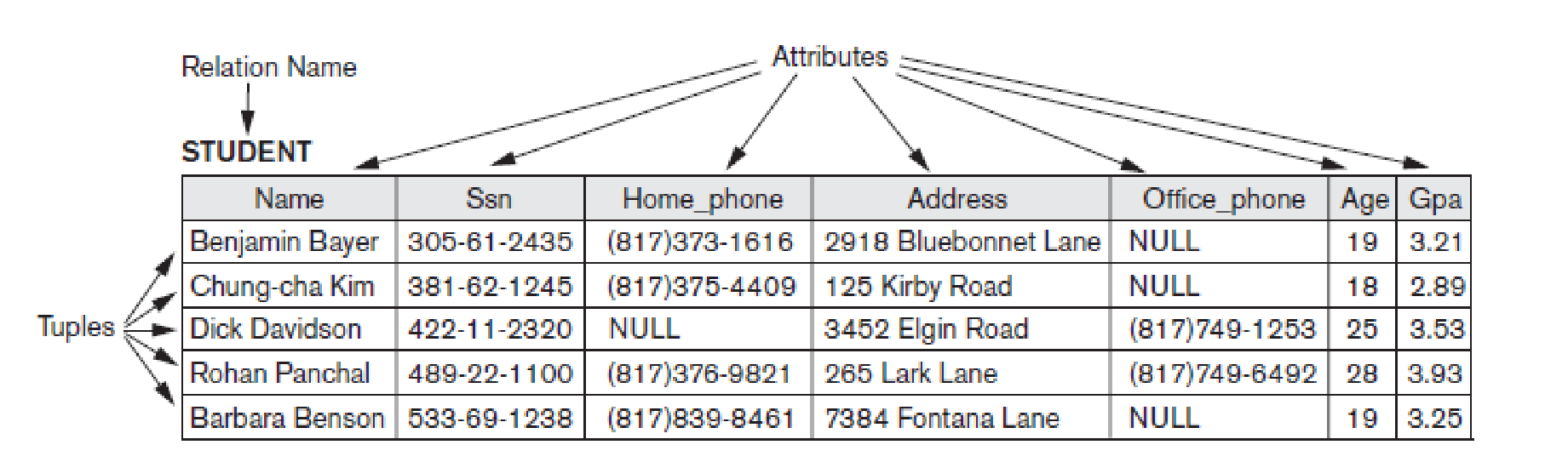
|  |  |  |
| --- | --- | --- |
| **RDBMS** | **Relational Model Theory** | **File Systems** |
| table | relation | file |
| row | tuple | record |
| column | attribute | field |

A **table** is a rectangular structure (matrix). It is a collection of organized values placed in columns and rows. A table can be represented in two ways:

* Relation schema: denoted by R(A1, A2, ...,An), is made up of a relation name R and a list of attributes A1, A2, ..., An, underlining the attributes that are the primary key.

STUDENT(Name, Ssn, Home\_phone, Address, Office\_phone, Age, Gpa)

* Representation expressed through a matrix.



Fundamentals of database systems / Ramez Elmasri

The **characteristics of tables/relations** are:

* In a table there are no repeated rows/tuples.

This means that no two rows/tuples can have the same combination of values for all their columns/attributes.

(Because in a set all the elements are different)

* The rows are not ordered within the table by any criteria.

(Because elements of a set have no order amount them)

* In a table there are no two columns with the same name.
* In a table cell, (a column of a row) may have no more than a simple value.

Regarding keys, it is defined the following concepts:

* **Identifier (or Superkey)**: is a subset of columns in the table that uniquely identifies each row in an isolated way. (A superkey has redundant attributes).
* **Key**: is an irreducible identifier. That is said that any subset of the columns that form the identifier cannot be an identifier as well. (A key must not have redundant attributes).

A table may have more than one key, in this case each of the keys is called **Candidate Key**.

In a table there is at least a candidate key (according to the set model there cannot be two exact rows).

* **Primary key (Primary Key)**: among all the candidate keys the designer has to choose one of them as the primary key by criteria unconnected with the relational model. It is a semantical integrity issue. The rest of the candidate keys will be renamed **unique keys.**

The primary key is of great importance in the model, because it is what allows the relationships among the tables.

* **Foreign Key**: is a set of columns from a table T2 whose values must match the primary key values from another table T1.

The Tables T1 and T2 can be the same table.

The structure of the relational model is extended with virtual tables called views.

A **view** is a derived table from other tables, which is represented in the system by its definition, but it does not have own stored data separated from the actual tables. A view offers a different perspective of data stored in tables.

We can specify SQL queries on a view in the same way we specify queries involving base tables. Some views can be updated, and others can’t.

Another structural element of the relational database systems are the indexes.

An **index** is an optional structure associated with columns in a table, which allows you to speed access to data in the table.

The indexes existence is completely “transparent” to programmers and users, but the DBMS always uses the existing indexes. So, the fact of defining indexes is a decision that concerns the database administrator to improve the system performance.

The indexes are logically and physically independent of the tables that are associated, hence you can create or delete indexes without affecting data.

If an index is defined on a column produce an improvement in the performance of the queries, but it also reduces the performance at the time of inserting, modifying, and deleting data because the indexes must be updated. Therefore, definition of indexes should be limited and correctly chosen to really improve the system performance.

It is advisable to define indexes on the columns that will be used to search for specific values. (for example, if you have frequently to access to a table by the column “Name”).

The DBMS Oracle automatically creates an index associated with each column defined as primary key and unique.

1. **Model Restrictions/Constraints.**

The relational model has a series of restrictions, that it is to say, structures or occurrences that are not permitted.

Restrictions can be divided in two categories:

* Inherent model-based constraints or implicit constraints that are the **integrity constraints**
* **Application-based constraints** or semantic constraints. These are restrictions on columns or rows that should be verified by the system to determine if they are valid in the information system.

The I**nherent model-based** constraints are:

- **Entity Integrity Constraint** states "No column that is part of the primary key can be NULL."

A NULL is not a value. It is a mark that is applied to a column to indicate that its value is not assigned yet or it is unknown.

- **Referential Integrity Constraint**: "A foreign key must have one of the values stored in the primary key of the table that is related to, or to be NULL."

**Rules for foreign keys**

Any state of the database that does not satisfy referential integrity is incorrect by definition.

There are a series of rules to avoid these incorrect states indicated at the time of the definition of the database, specifying what the system should make in the operation of deleting or updating that violate the referential integrity.

These are the following possibilities:

* **RESTRICT**: it is not allowed to delete rows or to modify the value of the primary key if there are foreign keys that contain its value.
* **CASCADE**: Deleting/updating the primary key causes the deleting/updating of the rows with the same values in a foreign key.
* **SET NULL**: Deleting/updating rows in table that contains the primary key causes the setting of the NULL value in the foreign keys with the same value.
* **SET DEFAULT**: Deleting/updating rows in table that contains the primary key causes the setting of the default value in the foreign keys with the same value.

- **Application-based constraint**: it is a predicate defined on columns or rows that must be verified by the system to determine if the information is valid. These restrictions can be stored in the data dictionary or be supported as triggers. A restriction of user must have a name, the predicate that must be satisfied and a response to the operation that attempts to violate the constraint.

(Eg: SALAR >= SALAR\_INTERPROFESSIONAL MINIMUM).

1. **Data Retrieval.**

Codd defined three mathematical systems included in the set theory, to manipulate the data that are structured according to the Relational model.

These languages are equivalent and have in common that they are not navigational languages (do not act on individual rows) but are languages of specification that act on sets of rows.

These languages are:

* Relational Algebra Oriented to tuples.
* Relational Calculus Oriented to tuples.
* Relational Calculus Oriented to Domains.

The DML (Data Definition Language) of the DBMS are based on one or more of these mathematician languages.

SQL, is based on the Relational Algebra and the Relational Calculus Oriented to tuples.