Databases. The Relational Data Model

LECT. DR. LIVIANA TUDOR

PETROLEUM-GAS UNIVERSITY OF PLOIESTI, ROMANIA

www.upg-ploiesti.ro

Basic concepts related to:
- database
- databases management systems
- relational data model

Designing a relational database consists of
- creating database objects and
- materializing them into conceptual diagram.

Databases. Databases Management Systems

A database

- can store information from everyday reality or a virtual universe

- a consistent structured data assembly without unnecessary redundancy that can be processed efficiently and concurrently by multiple users (Popescu, 2001; Tudor, 2011)

- a set of data organized, structured, stored in conditions of minimum redundancy, controlled and accessible to more users (Lungu, Bodea, Badescu, Ionita, 1995; Lungu, 2005; Tudor, 2011).

A database management system (DBMS)

- is a set of programs that allow creating of a database, updating and querying it (Bancilhon, 1988)

- is a software package that enables the following operations (Tudor, 2011):
  - creating database objects;
  - data entry and validation in database;
  - manipulation and data protection;
  - DB maintenance (backup, recovery, conversion, compacting etc).

Relational databases:

- Oracle
- Microsoft (Access, SQL Server)
- IBM (DB2, Informix)
- SAP (Sybase SQL)
- MySQL, PostgresSQL (freeware)

The Relational Data Model

- was defined and developed by Dr. E. F. Codd after 1970, being based on the mathematical theory of relations (Codd, 1970)

- uses simple data structures (tables or relations)

- characterized by integrity constraints, a lot of operators applicable to relations, for defining, manipulation and managing data

- has a relational language for describing and processing data, ensuring control of semantic data integrity and optimizing queries (query language)

Very popular query languages are:

- SQL (Structured Query Language) (e.g. Oracle, MySQL etc.)

- QBE (Query by Example) (e.g. Borland Paradox, Microsoft Access)

QBE was created by Moshe Zloof at IBM in the 1970s in parallel to SQL's development.
It is a graphical query language where users can input commands into a table like conditions and example elements

- Datalog (Petrescu, 2014)

Datalag is a query language that contains the logic rules "if-then". Prolog style syntax is commonly used.

Organizing Relational Data

Formal	Usually	Physically
Relation	Table	File
Tuple	Line	Record
Attribute	Column	Field
Domain	Data type	Data type

Table 1. Notations used in the relational model

The relational scheme for an R relation is marked $R(A_1, A_2, \dots, A_n)$, where $A_1, A_2, \dots, A_n$ are attributes (columns).

A *null value* is an unknown attribute or one that cannot be defined.

Integrity rules

= are conditions that data from the database must satisfy

Minimal integrity constraints are defined relative to the notion “key of a relation”.

Primary key in a table:
= a column (or a combination of columns) that contains not null values and uniques.

Example: Consider the table PERSONS with the following relational schema:

PERSONS (PIN, NAME, ADDRESS, CITY, BIRTHDAY, PHONE)

where PIN column is personal identification number and may be chosen primary key because it contains unique values and different of NULL

PIN	NAME	ADDRESS	CITY	BIRTHDAY	PHONE
1901240127640	Steven	15-80 Victoria Street	Sydney	1990-12-24	0745296014
2890819422782	Julia	148 The Oxford Science Park	Oxford	1989-08-19	0728105893
2011017278137	Clara	1297 Via Cola di Rie	Roma	2001-10-17	0739296018

Table 2. Table PERSONS

FOREIGN KEY:

Let it be two tables R1 and R2 with

- relational schemes R_1 (A, B) and R_2 (C, A), where
- A is the primary key in R_1, C - primary key in R_2 and
- A - secondary key in R_2 (fig. 1.1).

Then A is foreign key (or external key) for R_2 (Popescu, 2001).

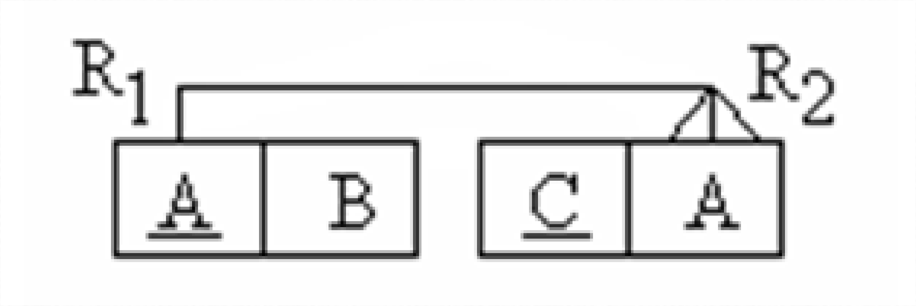

Fig. 1. *Example of foreign key in table R2*

Example: Consider an application that manages tables COMPANIES, ORDERS and SUPPLIERS with relational schemes below. It is required to achieve the database schema.

COMPANIES (Name, Address, Payment balance)

ORDERS (Product, Quantity, Name)

SUPPLIERS (Code, Product, Price)

COMPANIES

Name	Address	Payment balance
N1	Ad1	+Pb1
N2	Ad2	0
N3	Ad3	-Pb3
N4	Ad4	-Pb4

ORDERS

Product	Quantity	Name
P1	Q1	N1
P2	Q2	N1
P3	Q3	N3
P4	Q4	N4
P5	Q5	N3
P6	Q6	N3

SUPPLIERS		
Code	Product	Price
C1	P1	S1
C1	P6	S2
C1	P5	S3
C2	P4	S4
C2	P7	S5

The database schema is the following:

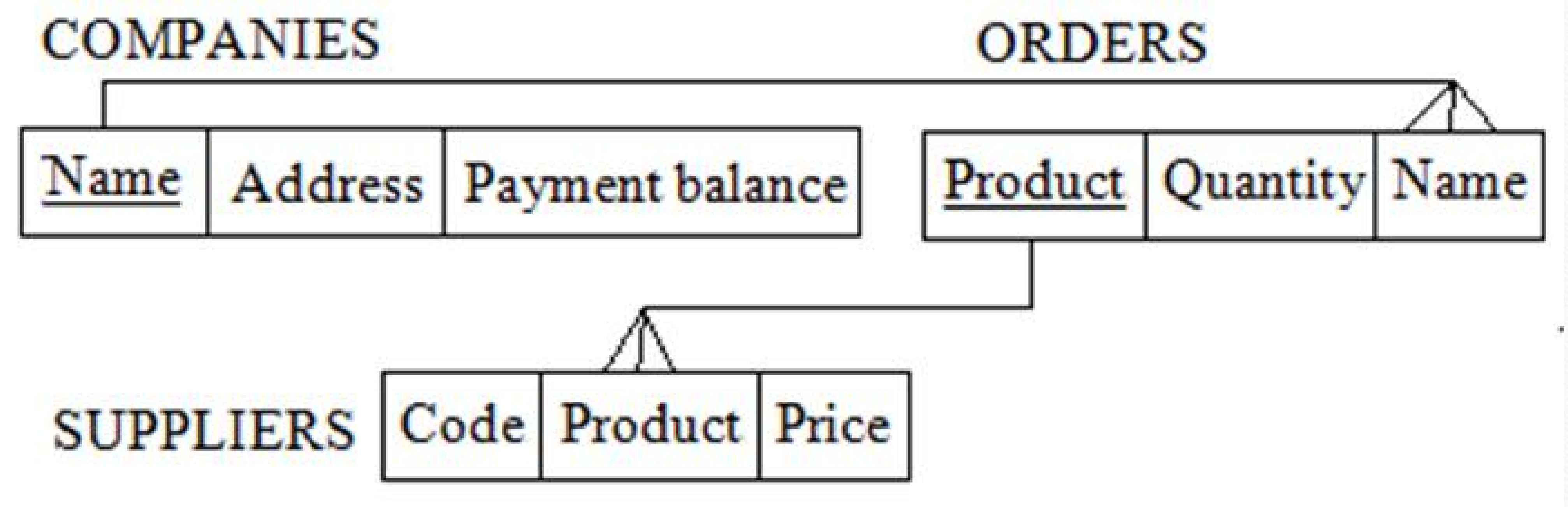

Fig. 2. *Primary key and Foreign key*

Structured Integrity Rules

There are three types of structural constraints (Popescu, 2001):

- the key
- entity and
- reference,

as the minimum set of integrity rules to be complied with by a relational database management system.

The relational model fulfills 3 structured integrity rules (Popescu, 2001):

- *Rule 1: uniqueness of key* - primary key must be unique and minimal
- *Rule 2: integrity of entity* - the primary key attributes must be different from the null value
- *Rule 3: integrity of reference* - A foreign key must be either entirely null or correspond to value of associated primary key

Relationships (associations) between tables (Lungu, 2005):

- **1-1** (one to one) - a value of the primary key column of a table may correspond to a single value of the primary key column in the second table
- **1-n** (one to many) - a value of the primary key column of the table "parent" corresponds to several of the foreign key values (table "child")
- **n-n** (many to many) - cannot be implemented practically and resorting to adding an additional table that will contain the primary keys for original tables and columns for the association between tables. The primary key of the intermediate table will be a compound key.

Relational algebra operators

- are defined by Codd
- represent the set of operations with the objects of relational database (Popescu, 2001; Petrescu, 2014; Tudor, 2007):

1. *SELECT* (*selection operator*)
2. *PROJECT (projection operator*)
3. *UNION (reunion*)
4. *INTERSECT*
5. *DIFFERENCE (difference operator*)
6. *DIFFERENCE (difference operator*)
7. *DIVISION (division operator*)
8. *JOIN (composition operator, junction*)

Codd's rules

In 1985, Codd published a set of 13 rules according to which a DBMS can be considered relational (Popescu, 2001; Tudor, 2007):

Rule 1 - *rule of data management* - A relational DBMS must be able to manage a database through its relational possibilities.

…

Rule 6 - *rule of query language* - must be at least one language for manipulating database (SQL).

*Rule 10 - Rule of logical data indepen*dence - Application programs should be transparent to any changes made to the data.

Normalization of relations

Normalization of conceptual relations allows to obtain "molecular" relations without losing anything from information, to eliminate redundancy and anomalies of updating data.

Normalization is a process of gradual improvement of the initial conceptual schemes of a relational database.

After each stage of improvement, relations in the database reach a certain degree of perfection, so they are in a certain *normal form*.

The need of normalization derives from problems caused by anomalies. There are abnormalities:

• when inserting data, • change/delete data

• redundancy in data (repeated occurrences)

• reconnection problems.

Normalization allows:

• suppression of logic redundancy

• avoidance of anomalies to update

• solve the reconnection problem

Mathematical theory of normalization → author E. F. Codd

- are built standard tables (called normal forms)

Normal forms of relations within a relational database are defined in relation to anomalies that may arise:

1. The first normal form (NF 1)

A relationship is in NF 1, if each attribute decomposing it corresponds to a indivisible value (atomic). A tuple can not contain repetitive attributes or groups of attributes.

ANF-1 algorithm that brings a relationship in NF 1 deletes the composed and repetitive attributes.

2. The normal form two (NF 2)

3. The normal form three (NF 3)

A relationship R is in NF 3 if and only if:
- Relation R is in NF 2
- Each attribute that is not a key (is not part of a key) depends directly on the primary key.

From a formal point of view, each attribute that is not a key (is not part of a key) is not transitively dependent of any key of R.

Example: A transitive dependency occurs in the following relation:

Book	Genre	Author	Author Nationality
Twenty Thousand Leagues Under the Sea	Science Fiction	Jules Verne	French
Journey to the Center of the Earth	Science Fiction	Jules Verne	French
Leaves of Grass	Poetry	Walt Whitman	American
Anna Karenina	Literary Fiction	Leo Tolstoy	Russian
A Confession	Religious Autobiography	Leo Tolstoy	Russian

The functional dependency {Book} → {Author Nationality} applies; that is, if we know the book, we know the author's nationality.

{Book} → {Author}

{Author} does not → {Book}

{Author} → {Author Nationality}

Therefore {Book} → {Author Nationality} is a transitive dependency.

4. *Normal Form 4 (NF 4)*

5. *Normal Form 5 (NF 5)*

Self-assessment test 1

1. What is the significance of concept "tuple"? Choose the right response.
   a. Row
   c. Column
2. What is the structural integrity rule which refers to the connection between the primary key and the external key? Set the actual response:
   a. The rule "referral integrity"
   b. The rule "entity integrity"
   c. The rule "key uniqueness"
3. What is the relational operator that extracts information from multiple tables linked via keys:
   a. JOIN
   b. SELECT
4. What type of key is formed by column / columns in a table that must contain unique values and the different from null:
   a. External key
   b. Primary key
   c. Foreign Keys

Bibliography

Bancilhon, F., Object Oriented Database Systems, Int. Symp. On Principles of Database Systems, Austin, 1988

Codd, E. F., A Relational Model for large Shared Data Banks, Communications of the ACM, Vol. 13, nr. 6, 1970

Codd, E.F., Further Normalization of the Data Base Relational Model (Data Base Systems), Computer Science Symposia Series 6, New York, 1971

Cyran, M., **Lane,** P., Polk, JP. et al., **Oracle® Database** Concepts, 2005

Lungu, I., Baze de date ORACLE. Limbajul SQL, Editura ASE, 2005

Lungu, I., **Bodea**, C., **Badescu**, G., **Ionita**, C., Baze de date. Organizare, proiectare şi implementare, Editura All Educational, 1995

Fotache, M., SQL Dialecte DB2, Oracle, Visual Foxpro, Editura Polirom, 2001

Oracle Database In-Memory Versus SAP Hana, 2014, http://www.oracle.com/technetwork

Petrescu, M., Operațiuni în modelul relațional. Introducere în algebra relațională, Curs Universitatea Politehnica Bucureşti, http://www.bazededate.org/ 2014

Popescu, I., Modelarea bazelor de date, Editura Tehnica, 2001

Tudor, N. L., INFORMATION SYSTEMS AND ORACLE DATABASE MANAGEMENT, MATRIX ROM Publishing House Bucharest, 2011

Tudor N. L., INFORMATION SYSTEMS AND DATABASE MANAGEMENT, Publisher Petroleum-Gas University of Ploiesti, 2007

Ullman, J. D., Principles of Database and Knowledge-Base Systems, Computer Science Press, New York, 1988