ICOM 6005 – Database Management Systems Design

Dr. Manuel Rodríguez Martínez
Electrical and Computer Engineering Department

Lecture 5 – September 4th, 2001

Readings

- Read
  - Chapter ? of old version of textbook
  - Chapter 4 of new version of texbook
  - Paper by E.F. Codd
    - A Relational Model of Data for Large Shared Data Banks
- Get:
  - ICOM6005-Manual #2 – from Engineering Reproduction Center

Some clarifications: Relational Model

- Relation model
  - Conceptual model for data
  - Data is modeled in tabular fashion
    - Rows is a record containing related information that describes an entity.
    - Columns are attributes of the entity (also called fields)
    - Row represents a group of related attributes that represent or give some information about the entity.
  - Relation schema – describes the characteristics of the relation
    - Relation name
    - List of attributes
      - Attribute name
      - Attribute domain
    - Domain
      - Name
      - Associated values

Attribute Domain

- The domain is the set of possible values that an attribute can assume.
  - Z – set of integers
  - A – set of dates after January 1st, 2000.
  - M – set of ages greater or equal than 18 years.
- In relational model, the domain is described by a domain name and data type.
  - Domain is equivalent to a data type in a programming language
    - Both describe legal values for a variable
    - Can be user-defined
      - Ex: M – set of ages greater or equal than 18 years.

Example: University Students

- Relation Schema:

  Students(sid:string, name:string,login:string, age:integer,gpa:integer);

- Relation name: Students
- Attributes: sid, name, login, age, gpa
- Domains: string, integer

Relation instance

- Relation instance – set of rows (also called tuples) that follow a relation schema R.
  - called relation for short.
- Technically, a relation should not have duplicates, but most commercial database management systems allow them.
- Example: Students(sid:string, name:string,login:string, age:integer,gpa:integer);

sid	name	login	age	gpa
1234	Manuel	manu@ece	18	3.50
6767	Jose	joe@ece	25	3.10
8901	Luz	luz@ece	21	4.00

Relation instance

- Order of the rows is irrelevant
- If the attributes in the schema are named, then the order is irrelevant – in theory
  - Most commercial systems assign a position to the fields
  - Example: Students(sid:string, name:string,login:string, age:integer,gpa:integer);
    - sid – column 1
    - name – column 2
    - login – column 3
    - age – column 4
    - gpa – column 5
  - You can access attributes (fields) by either name or position

Relational schema notation

- Given a set of Domain D1, D2, …, Dn a relation R over these domains is a subset of D1xD2x…Dn.
- R is specified as:
  - R = {a1:D2,a2:D2,…,an:Dn| where a1$\in$ D1, a2 $\in$ D2, …, an $\in$ Dn.
  - Sometimes, a relation R is described simply by schema as
    - R(a1:D2, a2:D2,…,an:Dn)
      - Lists the attribute names and their corresponding domain
  - If each domain is well-known or understood from the context of a document, then the relation is simply describe by the attribute names
    - R(a1, a2, …, an)
      - Merely indicates the attribute names in the

Keys

- Candidate keys – a set of attributes that uniquely identifies a tuple r in a relation R.
  - Just call it a key.
  - Two tuples cannot have the same key
  - In commercial DBMS this must be enforced by a constrain
    - Remember that commercial systems allow duplicate tuples
    - A constrain is a restriction on the values of one or more attributes.
  - If an attribute is remove from a candidate key, it is no longer a key.
- Super keys – set of attributes that contain a subset of attributes which is a candidate key
- Primary keys – candidate key selected by a database designer to be the key of a relation R.

Examples on keys

sid	name	login	age	gpa
1234	Manuel	manu@ece	18	3.50
6767	Jose	joe@ece	25	3.10
8901	Luz	luz@ece	21	4.00

- Candidate keys – {sid}, {login,age}
- Super Keys – {sig, name}, {sid, gpa}, {sid, login, age}
- Primary key – {sid}
- Non-keys – {login}, {age}, {name, login}

Some issues on keys

- Every relation has a candidate keys – in theory
  - Relational DBMS might allow relation with no keys
- A relation R might have multiple candidate keys
  - Example: Students can have either {sid} or {login,age}
- Primary key is chosen by DB designer. Usually is the candidate key that makes the most sense, meaning that is
  - Simple – as few attributes as possible.
    - Student ID – sid -> just one attribute
  - Intuitive – attributes naturally identify a tuple.
    - Student ID – sid -> natural way to identify a student

Relational Algebra

- Query language – specialized language to ask questions to the database.
  - These are normally called “queries”
- Relational algebra expresses queries as operations to be executed on the data (input relation).
  - Procedural expressions in the sense that they specify operation to be applied to an input relation
  - The do not specify how to implement these expressions, so they are not a procedural language.
- Relational algebra expression are specified over the schema of a relation, but are applied to an instance.
  - Relation names and attributes appear in the expressions.

Relational Algebra (cont.)

- In a relational algebra expression:
  - Input – is a relation R which can either be:
    - A base relation that exists as an instance in the database.
    - A temporary relation that is the result of another relational algebra expression.
      - Composition of relational algebra expressions
  - Output – a new relation R different from the base relations that are stored in the database (different from the existing instances).
- Relational algebra expressions can be the input to other expression.
- A relation is always the output of any relational algebra expression.

University Database Schema

- Students(sid:string, name:string,login:string,
  age:integer,gpa:integer);
- Courses(cid:string, section:integer, name:string,
  room:integer);

- Registration(cid:string, section:integer, sid:string);

- Key fields are underlined.

Relational Instance: Students

sid	name	login	age	gpa
9291	Mary	mary@ece	30	4.00
1234	Manuel	manu@ece	18	3.50
6767	Jose	joe@ece	25	3.10
3229	Bobby	bob@ece	19	3.90
9987	Rafael	rafy@ece	29	2.75
8901	Luz	luz@ece	21	4.00

Information about enrolled students

Relational Instance: Courses

cid	section	name	room
8010	1	English I	333
8010	2	English I	333
8010	3	English I	333
8074	1	Databases	123
8074	2	Databases	124
8075	1	Electronics	230
8087	1	Calculus	276

Information about courses being offered

Relational Instance: Registration

cid	section	sid
8074	1	1234
8087	1	1234
8010	1	6767
8010	1	8901
8010	2	3229
8075	1	9291
8087	1	1234

Information about student enrollment

Selection Operation

- Extracts a set of tuples from a relation R based on a boolean condition c.
  - Results is a set new relation with the same schema as input relation R.
  - Selection condition c is a predicate that restricts the tuples that are in the result.
    - Tuples must “pass” the condition to be in the selection result.
- Formalism:

  Given a relation R, the selection operation:

$$S = s_{c}(R)$$

  is the defined as S = { t | t $\in$ R and c(t) = true}

Selection Example 1

- Get all students with a gpa greater than 3.50.
- Relation Algebra Expression:

$$s_{gpa>3.50}(Students)$$

- Result: $R = s_{gpa>3.50}(Students)$

sid	name	login	age	gpa
9291	Mary	mary@ece	30	4.00
3229	Bobby	bob@ece	19	3.90
8901	Luz	luz@ece	21	4.00

Selection Example 2

- Get the course information for English I
- Relational Algebra Expression:

$$s_{name="English\ I"}(Courses)$$

- Result: $R = s_{name="English\ I"}(Courses)$

cid	section	name	room
8010	1	English I	333
8010	2	English I	333
8010	3	English I	333