

## Chapter 5

The Relational Data Model  
and Relational Database Constraints

## \* Data Models

- A set of concepts to describe the structure of a database, and certain constraints that the database should obey.
- Data model is a collection of conceptual tools for describing data, data relationships, data semantics, etc.

## \* Relational Model Concepts

- The relational model represents data and relationships among data by a collection of tables known as relations.
- A relation may be defined in multiple ways.
  - A relation may be regarded as a set of tuples (rows).
  - A relation may be thought of as a set of columns.
- The Schema of a Relation:  $R(A_1, A_2, \dots, A_n)$   
Relation Schema  $R$  is defined over attributes  $A_1, A_2, \dots, A_n$ .
- A tuple is an ordered set of values.
- Each row has a value of an item or set of items that uniquely identifies that tuple in table.
- Columns in a table are also called attributes of the relation.

**Tuples** : The rows of a relation are known as tuples

**Attributes** : The columns of a relation are known as attributes

**Degree** : The number of attributes in a relation

**Cardinality** : The number of rows in a relation



## \* Domain

- A domain is a set of atomic values.
- Domain is a unique set of values permitted for an attribute in a table.
- A data type or format is also specified for each domain.
- The relation is formed over a cartesian product of the sets, each set has values from a domain; that domain is used in a specific role which is conveyed by attribute name.
- Represented as  $\text{dom}(A_i)$  [ $A_i$  is an attribute].

→ Formally,

Given  $R(A_1, A_2, \dots, A_n)$

$$R \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \text{dom}(A_3) \dots \times \text{dom}(A_n)$$

$R$ : Schema of the relation

$r$  of  $R$ : a specific value or population of  $R$ .

$R$  is also called the intension of a relation.

$r$  is also called the extension of a relation.

- Relation is formed over the cartesian product of the sets; each set has values from a domain.

- The total number of values or cardinality, in domain  $D$  by  $|D|$ , the total no. of tuples in the Cartesian product is  $| \text{dom}(A_1) | \times | \text{dom}(A_2) | \times \dots \times | \text{dom}(A_n) |$



### \* Characteristics of Relations

→ Ordering of tuples in a relation  $R(R)$

The tuples are not considered to be ordered.

→ Ordering of values within a tuple.

Attributes in  $R(A_1, A_2, \dots, A_n)$  and the values in

$t = \langle v_1, v_2, \dots, v_n \rangle$  to be ordered.

→ Values and NULLs in the Tuples.

All values in a tuple are considered atomic.

A special null value is used to represent values that are unknown, unavailable or undefined.

### \* Notations

→ We refer to component values of a tuple  $t$  by

$t[A_i] = v_i$  (the value of attribute  $A_i$  for tuple

→  $t[A_u, A_v, \dots, A_w]$  refers to the subtuple of  $t$  containing the values of attributes  $A_u, A_v, \dots, A_w$ , resp.

### \* Relational Model Constraints

→ Constraint are conditions that must hold on all valid relation instances.

→ Constraints on database can be divided into 3 main categories:

- Inherent model-based or implicit constraints

Constraints that are inherent in the data model.

Example: In a relational model, no 2 tuples are duplicate.

- Explicit Constraints

Schema based constraints, specified by DDL.

(domain, unique, integrity, etc.)



- Application or semantic constraints. Constraints that cannot be directly expressed in the schemas, so needs to be enforced by application programs using assertions.

### \* Schema-based Constraints

- Domain constraint
- Key constraint
- Constraints on NULL
- Entity integrity constraints
- Referential integrity constraints

#### → Domain Constraint

Domain constraints specify that within each tuple, the value of each attribute  $A$  must be an atomic value from the domain  $\text{dom}(A)$ .

#### → Key Constraints

- Super key of  $R$  : A set of attributes  $SK$  of  $R$  such that no 2 tuples in any valid relation instance  $r(R)$  will have the same value for  $SK$ . That is, for any distinct tuples  $t_1$  and  $t_2$  in  $r(R)$ ,  $t_1[SK] \neq t_2[SK]$ .

\* Super key is a combination of all possible attributes which can uniquely identify 2 tuples in a table.

\* Super set of any candidate key is Super key.

- Candidate Key: All attribute combinations inside a relation that can serve as primary key are candidate keys. A superkey that does not contain a subset of attributes that is itself superkey



- Primary Key : A candidate key selected to uniquely identify tuples within the relation. Cannot contain null values.
- Secondary Key : An attribute (or combination of attributes) used strictly for data retrieval.
- Foreign Key : An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

→ A relational database schema  $S$  is a set of relation schemas  $S = \{R_1, R_2, \dots, R_m\}$  and a set of integrity constraints  $IC$ .

→ A database state that does not obey all the integrity constraints is called an invalid state.

→ A state that satisfies all the constraints in the defined set of integrity constraints  $IC$  is called a valid state.

→ Entity Integrity Constraint

- The entity integrity constraint states that no primary key value can be NULL. This is because primary key values are used to identify the individual tuples.

$t[PK] \neq \text{null}$  for any tuple  $t$  in  $r(R)$ .

- Other attributes of  $R$  may be similarly constrained to disallow null values, even though they are not members of the primary key.



## → Referential Integrity Constraint

- Referential Integrity constraint is specified b/w 2 relations and used to maintain the consistency among tuples in the 2 relations.
- Used to specify relationship among: the referencing relation and the referenced relation.
- Tuples in the referencing relation  $R_1$  have attributes FK (Foreign Key) that reference the primary key attributes PK of the referenced relation  $R_2$ .

A tuple  $t_1$  in  $R_1$  is said to reference a tuple  $t_2$  in  $R_2$  if  $t_1[FK] = t_2[PK]$ .

- The value in the FK column of the referencing relation  $R_1$  can be either :
  - i) A value of an existing primary key value of the corresponding primary key PK in the referenced relation  $R_2$  or

ii) a null

If null, then the FK in  $R_1$  should not be a part of its own primary key.

## \* Other Types of Constraints.

- Semantic Integrity constraints : based on application semantics and cannot be expressed by the model.
- Functional dependency constraints : establishes a functional relationship among 2 sets of attributes  $X$  and  $Y$ . This constraint specifies that the value of  $X$  determines a unique value of  $Y$  in all states of a relation. Denoted as  $X \rightarrow Y$ .

→ State constraints

→ Transition constraints



## \* Violation of constraints in relational database

### → Insert

On inserting the tuples in the relation, it may cause violation of the constraints:

#### i) Domain constraint

It gets violated when value of the attribute does not appear in corresponding domain or if the datatype is not appropriate.

#### ii) Entity Integrity Constraint

On inserting NULL values to any part of the primary key of a new tuple in the relation can cause violation of Entity Integrity constraint.

#### iii) Key constraints

On inserting a value in the new tuple of a relation which is already existing in another tuple of the same relation, can cause violation.

#### iv) Referential Integrity

On inserting a value in the foreign key of  $R_1$ , for which there is no corresponding value in the Primary key which is referred in  $R_2$ , in such cases Referential Integrity constraint is violated.

### → Delete

Delete operation can only violate referential integrity constraint.

It causes violation only if the tuple in  $R_1$  is deleted which is referenced by FK from other tuples of  $R_2$  in the db, if such deletion takes place then the values in the tuple of the FK in  $R_2$  will become empty, Hence Referential Integrity violated.

• Solution = Restrict, Cascade, Set null or defaults.

→ Update

i) Domain constraint

while updating the attribute.

ii) NOT NULL constraint

can be violated on being modified

iii) Key constraint

while updating PK or FK.

iv) Referential Integrity Constraint

While updating the foreign key.