Chapter 5, The Relational Data Model and Relational Database Constraints

5.1 Relational Model Concepts

- Represent the database as a collection of relations.
- Relation is a table of values.
- Each row represents an entity or relationship.
- table ↔ relation; row ↔ tuple; column header ↔ attribute.

5.1.1 Domains, Attributes, Tuples, and Relations

- A relation schema \( R(A_1, A_2, \ldots, A_n) \) with corresponding set of domains \((D_1, D_2, \ldots, D_n)\), where \(\text{dom}(A_i) = D_i\) for \(1 \leq i \leq n\). The degree of \(R\) is \(n\) (the # of attributes).

- A relation (or relation state) \(r(R) = \{t_1, t_2, \ldots, t_m\}\) is a set of n-tuples, where each \(t = < v_1, v_2, \ldots, v_n >\) and each \(v_i \in \text{dom}(A_i) \cup \text{NULL}\).
  
  \(t[A_i]:\) the \(i^{th}\) value in the tuple \(t\).

- \(r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \ldots \times \text{dom}(A_n))\)
  
  The under-lining expression above is all possible combination of values for a relation with degree \(n\).

5.1.2 Characteristics of Relations

- There is no ordering of tuples in a relation.

- Ordering of values (attributes) within a tuple depends on the formal definition.

- First Normal Form: each value in a tuple is \textbf{atomic} (no composite and multivalued attributes).
  
  Composite → simple component attributes.
  
  Multivalued → separate relations.
• Entity type and relationship type are represented as relations in relational model.
  e.g. STUDENT relation of Figure 5.1 (Fig 7.1 on e3) and a relation schema MAJORS
  (studentSSN, DepartmentCode)

5.1.3 Relational Model Notation

• $R(A_1, A_2, \ldots, A_n)$: a relation schema $R$ of degree $n$.

• $t = < v_1, v_2, \ldots, v_n >$: an n-tuple $t$ in $r(R)$.

• $t[A_i]$ or $t.A_i$: value $v_i$ in $t$ for attribute $A_i$.

• $t[A_u, A_w, \ldots, A_z]$ or $t.(A_u, A_w, \ldots, A_z)$: subtuple of values $< v_u, v_w, \ldots, v_z >$ from $t$
  for attributes $A_u, A_w, \ldots, A_z$.

• $Q, R, S$: relation names.

• $q, r, s$: relation states.

• $t, u, v$: tuples.

• $R.A$: attribute $A$ in a relation schema $R$.

5.2 Relational Model Constraints and Relational
Database Schemas

5.2.1 Domain Constraints

• Domain Constraint: all $t[A_i]$ must be atomic and belong to $\text{dom}(A_i) \cup \text{NULL}$.

5.2.2 Key Constraints and Constraints on NULL

• A superkey $SK$ of a schema $R$ is a subset of attributes that $t_1[SK] \neq t_2[SK]$.

• Default superkey: the set of all attributes.

• A key is a minimal superkey, i.e., you can not remove any attribute from the key and
  still make it as a superkey.
If there are several keys in a relation schema, each of the keys is called a candidate key. See Figure 5.4 (Fig 7.4 on e3).

Pick one from all candidate keys as the primary key.
Criterion for choosing a primary key: smallest # of attributes.

5.2.3 Relational Databases and Relational Database Schemas

- A relational database schema \( S = \{R_1, R_2, \ldots, R_m\} \) and a set of integrity constraints.

- A relational database state \( DB = \{r_1, r_2, \ldots, r_m\} \) for each \( r_i(R_i) \) such that each \( r_i \) satisfies integrity constraints.

- See Figure 5.5 and 5.6 (Fig 7.5 and 7.6 on e3).

5.2.4 Entity Integrity, Referential Integrity, and Foreign Keys

- Entity integrity constraint: No primary key value can be NULL.

- Referential integrity constraint: A tuple in one relation that refers to another relation must refer to an existing tuple.

- A set of attributes \( FK = \{A_u, A_v, \ldots, A_z\} \) is a foreign key of \( R_1 \) that references \( R_2 \) if
  
  (i) \( \text{dom}(A_u) = \text{dom}(A_v), \text{dom}(A_v) = \text{dom}(A_w), \ldots, \text{dom}(A_z) = \text{dom}(A_{z'}) \), where \( PK = \{A_u', A_v', \ldots, A_z'\} \) in \( R_2 \).

  (ii) For any \( t_1 \) in \( R_1 \), \( \exists \) a \( t_2 \) in \( R_2 \) such that \( t_1[FK] = t_2[PK] \) or \( t_1[FK] = \langle NULL, NULL, \ldots, NULL \rangle \)

These two conditions of foreign key specify the referential integrity constraint formally. We call \( R_1 \) referencing relation, \( R_2 \) referenced relation.

- A foreign key can refer to its own relation. See Figure 5.5 (Fig 7.5 on e3), the SUPER-SSN in EMPLOYEE.
• The diagrammatic representation of the referential integrity constraint: see Figure 5.7 (Fig 7.7 on e3).

5.3 Update Operations and Dealing with Constraint Violations

5.3.1 The Insert Operation

• INSERT a tuple \( t = < v_1, v_2, \ldots, v_n > \) to a relation \( R \) with degree \( n \).
  
  - It may violate domain, key, entity integrity and referential integrity constraints (see examples in pp. 162).
  
  - If there is a violation, either reject the insertion or try to correct the reason for rejection.

5.3.2 The Delete Operation

• DELETE a tuple (tuples) \( t \) from \( R \) if \( t \) satisfies the (selection) condition.
  
  - (selection) condition: \(< \text{Clause} > < \text{Boolean OP} > < \text{Clause} > < \text{Boolean OP} > \ldots\)
  
  - Clause: \(< \text{Attribute Name} > < \text{Comparison OP} > < \text{Const Value} > \) or \(< \text{Attribute Name} > < \text{Comparison OP} > < \text{Attribute Name} > \)
  
  - See examples on page 163 of the book.
  
  - Deletion may violate referential integrity constraint.
  
  - If there is a violation, three options: rejection, cascade the deletion, or modify the referencing attribute values.

5.3.3 The Update Operation

• UPDATE the values of some attributes in a tuple (tuples) \( t \) in \( R \) if \( t \) satisfies the (selection) condition.

• Updating an attribute that is
- a primary key: similar to delete a tuple, then insert another tuple.
- a foreign key: may violate referential integrity or domain constraints.
- otherwise, may violate domain constraint.
- See examples on page 164 on the book.

Exercise 5.11 on page 166 of the book.