The Relational Data Model and Relational Database Constraints

Part 3

Relational Model Constraints and Relational Database Schemas

Part 2

Relational Databases and Relational Database Schemas

- A **relational database schema** *S* is a set of relation schemas $S = \{R_1, R_2, ..., R_m\}$ and a set of integrity constraints IC.
- A **relational database state** *DB* of *S* is a set of relation states $DB = \{r_1, r_2, ..., r_m\}$ such that each r_i is a state of R_i and such that the r_i relation states satisfy the integrity constraints specified in IC.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

- When we refer to a relational database, we implicitly include both its schema and its current state.
- A database state that does not obey all the integrity constraints is called an **invalid state**, and a state that satisfies all the constraints in the defined set of integrity constraints IC is called a **valid state**.

Integrity, Referential Integrity, and Foreign Keys

- The **entity integrity constraint** states that no primary key value can be NULL.
- This is because the primary key value is used to identify individual tuples in a relation.
- Having NULL values for the primary key implies that we cannot identify some tuples.

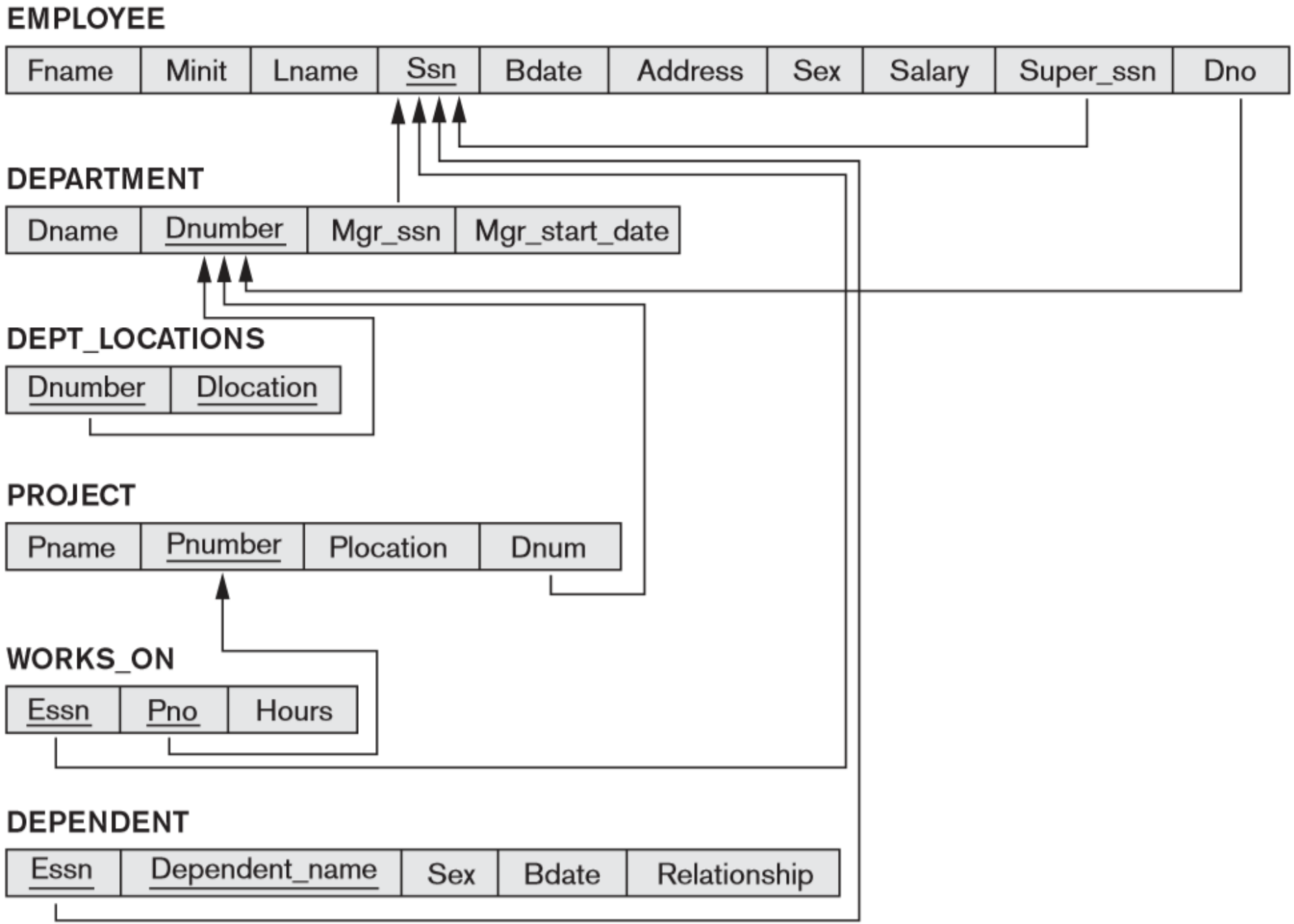
- The **referential integrity constraint** is specified between two relations and is used to maintain the consistency among tuples in the two relations.
- Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an *existing tuple* in that relation.

- A set of attributes FK in relation schema R_1 is a **foreign key** of R_1 that references relation R_2 if it satisfies the following rules:
  1. The attributes in FK have the same domain(s) as the primary key attributes PK of R_2; the attributes FK are said to **reference** or **refer to** the relation R_2.
  2. A value of FK in a tuple t_1 of the current state $r_1(R_1)$ either occurs as a value of PK for some tuple t_2 in the current state $r_2(R_2)$ *or is NULL*. In the former case, we have t_1[FK] = t_2[PK], and we say that the tuple t_1 references or refers to the tuple t_2.
- In this definition, R_1 is called the **referencing relation** and R_2 is the **referenced relation**.
- If these two conditions hold, a **referential integrity constraint** from R_1 to R_2 is said to hold.

- Referential integrity constraints typically arise from the *relationships among the entities* represented by the relation schemas.

- Notice that a foreign key can *refer to its own relation*.

- We can *diagrammatically display referential integrity constraints* by drawing a directed arc from each foreign key to the relation it references.


EMPLOYEE
Fname
Minit
Lname
Ssn
Bdate
Address
Sex
Salary
Super_ssn
Dno
DEPARTMENT
Dname
Dnumber
Mgr_ssn
Mgr_start_date
DEPT_LOCATIONS
Dnumber
Dlocation
PROJECT
Pname
Pnumber
Plocation
Dnum
WORKS_ON
Essn
Pno
Hours
DEPENDENT
Essn
Dependent_name
Sex
Bdate
Relationship

Update Operations, Transactions, and Dealing with Constraint Violations

- There are three basic operations that can change the states of relations in the database: Insert, Delete, and Update (or Modify).
- Whenever these operations are applied, the integrity constraints specified on the relational database schema should not be violated.

The Insert Operation

- The **Insert** operation provides a list of attribute values for a new tuple t that is to be inserted into a relation *R*.
- Insert can violate any of the four types of constraints.
- Domain constraints can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type.
- Key constraints can be violated if a key value in the new tuple t already exists in another tuple in the relation *r*(*R*).
- Entity integrity can be violated if any part of the primary key of the new tuple t is NULL.
- Referential integrity can be violated if the value of any foreign key in *t* refers to a tuple that does not exist in the referenced relation.

- If an insertion violates one or more constraints, the default option is to *reject the insertion*.

The Delete Operation

- The **Delete** operation can violate only referential integrity.
- This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database.

- Several options are available if a deletion operation causes a violation.
- The first option, called **restrict**, is to *reject the deletion*.
- The second option, called **cascade**, is to *attempt to cascade (or propagate) the deletion* by deleting tuples that reference the tuple that is being deleted.
- A third option, called **set null** or **set default**, is to *modify the referencing attribute values* that cause the violation; each such value is either set to NULL or changed to reference another default valid tuple.

The Update Operation

- The **Update** (or **Modify**) operation is used to change the values of one or more attributes in a tuple (or tuples) of some relation *R*.
- It is necessary to specify a condition on the attributes of the relation to select the tuple (or tuples) to be modified.

- Updating an attribute that is *neither part of a primary key nor of a foreign key* usually causes no problems; the DBMS need only check to confirm that the new value is of the correct data type and domain.
- Modifying a primary key value is similar to deleting one tuple and inserting another in its place because we use the primary key to identify tuples.
- If a foreign key attribute is modified, the DBMS must make sure that the new value refers to an existing tuple in the referenced relation (or is set to NULL).

The Transaction Concept

- A database application program running against a relational database typically executes one or more *transactions*.
- A **transaction** is an executing program that includes some database operations, such as reading from the database, or applying insertions, deletions, or updates to the database.
- At the end of the transaction, it must leave the database in a valid or consistent state that satisfies all the constraints specified on the database schema.

- ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability