COS221
L08 - Relational Model and Constraints
(Chapter 3 in Edition 6 and Chapter 5 in Edition 7)

Linda Marshall

10 March 2023

Relational Model

- First introduced by Codd in 1970
- Became popular because of its simplicity and mathematical foundation
  - Basic building block - a *table of values*
  - Theoretical basis - set theory
- First commercial RDBMSs became available in the 1980's
  - DB2 from IBM
  - Oracle
  - Sybase (now from SAP)
  - SQLServer and Access from Microsoft
  - MySQL, PostgreSQL and MariaDB, all open source

Relational Model Concepts

- A database is represented as a collection of relations
- A relation can be seen as a table (flat file) of values (records)
  - each row (*tuple* in relational database terminology) of a table (*relation*) is collection of related values and typically corresponds to a real-word entity or relationship
  - the table name and column names (*attributes*) are used to interpret the meaning of the values in each row
  - the values in the columns are described by types (*domains*)

Relational Model Concepts - *Relation schema*

A *relation schema*, R, is made up of a relation name, R, and a list of attributes: $A_1, A_2, ..., A_n$ and denoted by $R(A_1, A_2, ..., A_n)$

- Each attribute, A_i, is the name of the role played in domain, D, in the relation schema R.
- The domain of A_i is denoted by $dom(A_i)$
- The *degree* (arity) of a relation is the number of attributes (n) of its relation schema

For example:

```
STUDENT(Name:string, SSn:string, Cell_number:string,
        Age:integer, Gpa:real)
```

Relational Model Concepts - *Relation*

An alternative representation of a relation is as a **predicate**.

- Values in a tuple are interpreted as values satisfying a predicate
- The relational model can then be manipulated by logic languages such as Prolog

The **closed world assumption** states that only true facts in the universe are those present within the extension (state) of the relation(s). This comes in useful when queries based on relational calculus are considered.

Relational Model Concepts - *Relation*

- A relation, r, of $R(A_1, A_2, ..., A_n)$, also denoted by $r(R)$, is the set of n-tuples, $r = \{t_1, t_2, ..., t_n\}$
- Each n-tuple, t, is an ordered list of n values $t =< v_1, v_2, ..., v_n >$
- Each v_i is an element $dom(A_i)$, or a special value NULL – that is, not known, not available, undefined or may not apply to the tuple
- The i^{th} value of tuple t, which corresponds to attribute A_i, is referred to as $t[A_i]$,$t.A_i$ or $t[i]$

Relational Model Concepts - *Relation*

- R, the relation schema, is referred to as the **relation intension**, while $r(R)$ – the relation state – is referred to as the **relation extension**
- A tuple can be seen as asserting a fact. Some relations represent facts about *entities* and others about *relationships*. For example, a relation:

  `MAJORS(Student_id, Department_code)`

  asserts that a students major in specific disciplines
- (E)ER conceptual models will be converted to relations where some relations represent the entities and others represents relations in the (E)ER-diagrams

Relational Model Concepts - *Domain*

- A domain, D, is a set of atomic values. That is, they are indivisible in the relational model.
  - Atomic implies that neither composite nor multivalued attributes are allowed in the relational model. This is the reason for the first normal form (1NF) assumption.
- A domain is defined by a name, data type and format.
  - Naming domains helps with the interpretation of the values
  - Data types are used to specify the type of the data values in the domain
  - Format ensures that the data, if need be, adheres to a specific format
- For example: The domain `Telephone_number` is a set of 10 digits. The data type of a domain is a character string of the form `(ddd) ddd-dddd`

Relational Model Concepts - *Domain*

- A relation $r(R)$ of degree n on domains $dom(A_1), dom(A_2), ..., dom(A_n)$ is a subset of the cartesian product of the domains that define R
  $r(R)$ $(dom(A_1) \times dom(A_2) \times ... \times dom(A_n))$
- The total number of tuples in the cartesian product will be: $|dom(A_1)| \times |dom(A_2)| \times ... \times |dom(A_n)|$, where $|D|$ denotes the cardinality of the domain

Relational Model Concepts - *Summary*

Relation Name

Attributes

STUDENT

Tuples

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1
The attributes and tuples of a relation STUDENT.

Relational Model Characteristics - Tuples

- A relation is defined as a set of tuples. This means there is no ordering of tuples.
- The relations below are therefore seen as identical

Relation Name

Attributes

STUDENT

Tuples

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1
The attributes and tuples of a relation STUDENT.

Figure 5.2
The relation STUDENT from Figure 5.1 with a different order of tuples.

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Relational Model Characteristics - Attributes

- The ordering of attributes is important. An n-tuple is an ordered list of n values. When manipulating n-tuples, the ordering of the corresponding attributes and its values need to be maintained
- If ordering of attributes is not defined as part of the relation definition, then a tuple can be defined as a set of $(< attribute >, < value >)$ pairs to preserve the mapping from R to D for each t_i. In this mapping, D is the union of domains of the attributes in R.

$t = <$ (Name, Dick Davidson),(Ssn, 422-11-2320),(Home_phone, NULL),(Address, 3452 Elgin Road),
(Office_phone, (817)749-1253),(Age, 25),(Gpa, 3.53)$>$

$t = <$ (Address, 3452 Elgin Road),(Name, Dick Davidson),(Ssn, 422-11-2320),(Age, 25),
(Office_phone, (817)749-1253),(Gpa, 3.53),(Home_phone, NULL)$>$

Figure 5.3
Two identical tuples when the order of attributes and values is not part of relation definition.

Relational Model Constraints

- Relational model constraints exist because there are constraints between the tuples of the relations of a database.
- Constraints are divided into the following categories:
  - Implicit or **inherent model-based** constraints
  - Explicit or **schema-based** constraints. These are directly expressed in schemas by specifying them in the DDL (Data Definition Language). These include: domain, key, NULL value, entity integrity and referential integrity constraints.
  - Semantic or **application-based** constraints, also referred to as business rules. The application programmes usually check these when performing updates.
  - **Dependency** constraints such as functional and multivalued dependencies. These are used to test the 'goodness' of the design.

Relational Model Constraints - Schema-based

The following schema-based constraints exist:

- Domain
- Key
- NULL values
- Entity integrity
- Referential integrity

Relational Model Constraints - Schema-based

- **Domain** - An attribute, A, must be an atomic value from the domain, $dom(A)$
- **Key**
  - As all elements of a set must be unique, all tuples in a relation must be unique/distinct.
  - If we can find a subset of attributes (SK, the superkey) of a relation so that for any two distinct tuples t_1 and t_2, in r of R, $t_1[SK]! = t_2[SK]$, then R is distinct.
  - A minimal superkey, or key, is defined as a set of attributes from which the removal of a single attribute no longer renders it unique.
  - A relation may have more than one key. If this is the case, then each of the potential keys is referred to as a **candidate key**. One of the candidate keys is designated the **primary key**. Only the primary keys are *underlined* when drawing a relational schema.
  - A key comprising of multiple attributes, requires all attributes together to provide uniqueness.

Relational Model Constraints - Schema-based

- **NULL values** - An attribute defined as NOT NULL, may not hold the NULL value.
- **Entity integrity** - No primary key value may be NULL.
- **Referential integrity** - Referential integrity constraints are specified between two relations and is used to maintain consistency between the relations.

Relational Model Constraints - Schema-based

- **Referential integrity** - *continued* - To formally specify referential integrity, the concept of a **foreign key** is needed. A set of attributes, FK in a relation schema R_2, is a foreign key of R_2 that references relation R_1 if it satisfies the following:
  - The attributes of FK have the same domain as the PK attributes of R_1. FK references or refers to R_1.
  - A value of FK in the tuple t_2 of the current state $r_2(R_2)$ either occurs as a value of PK for some tuple t_1 in the current state $r_1(R_1)$ or is NULL. If not NULL, we have $t_2[FK] = t_1[PK]$ and t_2 references or refers to tuple t_1.

  R_2 is referred to as the *referencing relation* and R_1 the *referenced relation*. If the conditions hold, a referential integrity constraint from R_2 to R_1 holds. That is, from the FK in R_2 to the PK in R_1.

Relational Database Schema

- 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).

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

Figure 5.5
Schema diagram for the COMPANY relational database schema.

Note: Attribute names are context specific. We could have had `Name` in both PROJECT and DEPARTMENT. `Dno`, `Dnumber` and `Dnum` all refer to the department number.

Relational Database Schema

- A relational database state (DB of S) is the set of relation states $DB = \{r_1, r_2, ..., r_m\}$.

Figure 5.6
One possible database state for the COMPANY relational database schema.

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

Relational Database Schema - Referential Integrity

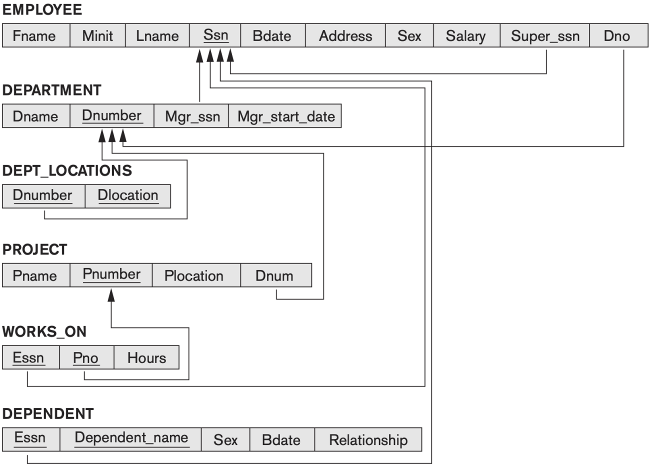

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Update operations and dealing with constraint violations

Update operations fall into two categories.

- **Retrieval** - makes use of Relational Algebra to query the database. The result relation is the answer to the user's query. This will be covered later.
- **Update** - Changes the state of the relations in the database. Three basic operations exist to change the state of a database.
  - Insert - Used to add one or more tuples into a relation
  - Delete - Used to delete tuples.
  - Update/Modify - Used to change values of some attributes of existing tuples.

Update operations and dealing with constraint violations - Insert

Provides a list of attribute values for a new tuple t. It can violate:

- Domain constraints - attribute value does not appear in the corresponding domain or not of the appropriate data type.
- Key constraints - key value in the new tuple already exists in another tuple in $r(R)$
- Entity integrity constraints - any part of the primary key in the new tuple is NULL
- Referential integrity constraints - the value of a foreign key refers to a tuple that does not exist in the referenced relation

Update operations and dealing with constraint violations - Delete

Delete only violates referential integrity constraint. The following options exist if delete causes a violation:

- Restrict - reject the deletion
- Cascade - propagate the deletion by deleting tuples that reference the the tuple being deleted
- Nullify - set the referencing attributes that cause the violation to NULL or change to reference another default valid tuple. If the referencing tuple is part of the primary key it cannot be set to NULL so as to not violate entity integrity

Update operations and dealing with constraint violations - Update/Modify

- Used to change the values of one or more attributes in the tuple(s) of some relation R
- A condition on attribute(s) of the relation needs to be specified to select the tuple(s) to be modified
- Updating an attribute that is neither part of a primary key or a foreign key causes no problems
- Modifying the primary key is similar to deleting the tuple and adding a new one with the modified values in its place.
- When modifying a foreign key, the value to which it is modified must exist in the relation being referred to, or must be set to NULL

Update operations and dealing with constraint violations

Figure 5.6
One possible database state for the COMPANY relational database schema.

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

Transactions and dealing with constraint violations

- A transaction is a program that manipulates - reads and/or updates - the database
- Once a transaction has completed, the database must be in a valid or consistent state that satisfies the constraints specified by the database schema
- In large online transaction processing (OLTP) systems, many transactions run in milliseconds and concurrently and therefore concurrency and recovery from failure needs to be managed