Course Contents

Unit-01: Database Concepts and Architecture	(4 Hrs.)
Unit-02: Data Modeling Using the Entity-Relational Model	(5 Hrs.)
Unit-03: The Relational Data Model and Relational Database Constraints	(5 Hrs.)
Unit-04: SQL	(10 Hrs.)
Unit-05: Relational Database Design	(7 Hrs.)
Unit-06: Transaction Processing and Concurrency Control, and Recovery	(8 Hrs.)
Unit-07: Database Recovery Techniques	(3 Hrs.)
Unit-08: NoSQL	(3 Hrs.)

Course Contents

Unit-03:The Relational Data Model and Relational Database Constraints

- Relational Model Concepts;
- Relational Model Constraints and Relational Database Schemas;
- Update Operations, Transactions, and Dealing with Constraint Violations;
- Basic Relational Algebra Operations

Course Contents

Unit-03:The Relational Data Model and Relational Database Constraints (5 Hrs.)

1. What is Relational Data model? How Relation is different than Flat file or Table (Explain the characteristics of Relation).
2. Explain the terms Domain, Attributes, Tuples, Records, Relation
3. What is Relation? Explain the mathematical representation of Relation.
4. What are Relational Model Constraints or simply constraints? Explain different types of constraints.
5. Explain the types of constraints violation in the database state during Database operation like Insert, Delete, and Update (or Modify).

The Relational Data Model and Relational Database Constraints

Introduction:

- The relational data model was first introduced by Ted Codd of IBM Research in 1970 and is a mathematical foundation of representing the data.
- The model uses the concept of a mathematical relation and have basic characteristics and constraints of the Relational Model.
- The first commercial implementations of the relational model became available in the early 1980s, such as the SQL/DS system on the MVS (Multiple Virtual Storage) operating system by IBM and the Oracle DBMS.
- Now, has been implemented in a large number of commercial systems, open source systems.
- Current popular commercial relational DBMSs (RDBMSs) include DB2 (from IBM), Oracle (from Oracle), Sybase DBMS (now from SAP), and SQLServer and Microsoft Access (from Microsoft).
- In addition, several open source systems, such as MySQL and PostgreSQL.

The Relational Data Model and Relational Database Constraints

Introduction:

Will will learn:

- Basic principles of the relational model of data, modeling concepts and notation of the relational model.
- Relational constraints that are considered an important part of the relational model and are automatically enforced in most relational DBMSs.
- Defines the update operations of the relational model, discusses how violations of integrity constraints are handled, and the concept of a transaction.

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

- The relational model represents the database as a collection of relations. Each relation resembles a table of values or also call flat file of records (flat-file). It is called a flat file because each record has a simple linear or flat structure.
- Flat file is similar to the basic relational model representation. However, there are important differences between relations and flat files. These rows in the table denote a real-world entity or relationship.

Relation Name

Attributes

STUDENT

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

Tuples

Relational Model Concepts :

- The table name and column names are used to help to interpret the meaning of the values in each row.

Relation Name

Attributes

Tuples

STUDENT

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

Relational Model Concepts:

- In the formal relational model terminology, a row is called a tuple, a column header is called an attribute, and the table is called a relation.
- The data type describing the types of values that can appear in each column is represented by a domain of possible values.
- We now define these terms- domain, tuple, attribute, and relation

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

- Domain: A domain is the set of legal values that can be assigned to an attribute. Each attribute in a database must have a well-defined domain. One goal of database developers is to provide data integrity, part of which means insuring that the value entered in each field of a table is consistent with its attribute domain.

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

Some examples of domains follow:

- Social_security_numbers: The set of valid nine-digit Social Security numbers. (unique identifier assigned to each person for employment, tax, and benefits purposes.)
- Names: The set of character strings that represent names of persons.
- Grade_point_averages: Possible values of computed grade point averages; each must be a real (floating-point) number between 0 and 4.
- Employee_ages: Possible ages of employees in a company; each must be an integer value between 15 and 80.
- Academic_department_names: The set of academic department names in a university, such as Computer Science, Economics, and Physics.
- Academic_department_codes: The set of academic department codes, such as ‘CS’, ‘ECON’, and ‘PHYS’.

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

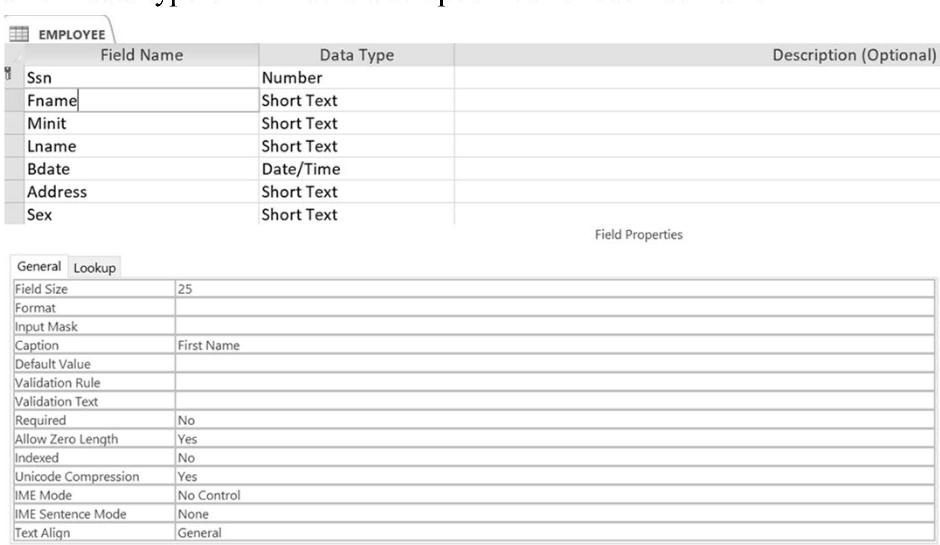
Domain: A data type or format is also specified for each domain.

- A **domain** has a logical definition:
  - Example: "USA_phone_numbers" are the set of 10 digit phone numbers valid in the U.S.
- A domain also has a data-type or a format defined for it.
  - The USA_phone_numbers may have a format: (ddd)ddd-dddd where each d is a decimal digit.
  - Dates have various formats such as year, month, date formatted as yyyy-mm-dd, or as dd mm,yyyy etc.
- The attribute name designates the role played by a domain in a relation:
  - Used to interpret the meaning of the data elements corresponding to that attribute
  - Example: The domain Date may be used to define two attributes named "Invoice-date" and "Payment-date" with different meanings

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

Domain: A data type or format is also specified for each domain.



EMPLOYEE

Field Name	Data Type	Description (Optional)
Ssn	Number	
Fname	Short Text	
Minit	Short Text	
Lname	Short Text	
Bdate	Date/Time	
Address	Short Text	
Sex	Short Text	

Field Properties

General | Lookup

Field Size	25
Format	
Input Mask	
Caption	First Name
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

Domain: A data type or format is also specified for each domain.

- A relation schema R, denoted by $R(A_1, A_2, \ldots, A_n)$, is made up of a relation name R and a list of attributes, $A_1, A_2, \ldots, A_n$. Each attribute A_i is the name of a role played by some domain D in the relation schema R. D is called the domain of Ai and is denoted by $dom(A_i)$.

- A relation schema is used to describe a relation; R is called the name of this relation. The degree (or arity) of a relation is the number of attributes n of its relation schema. A relation of degree seven, which stores information about university students, would contain seven attributes describing each student as follows: STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)
- Using the data type of each attribute, the definition is written as: STUDENT(Name: string, Ssn: string, Home_phone: string, Address: string, Office_phone: string, Age: integer, Gpa: real)

The Relational Data Model and Relational Database Constraints

Relational Model Concepts:

Attributes: The main difference between attribute and domain is that an attribute is a property that represents an entity while a domain is a collection of values that can be assigned to an attribute.

ATTRIBUTE	DOMAIN
A descriptive property which is owned by each entity of an entity set	The set of values allowed for an attribute
Helps to describe an entity	Helps to define the range of values that suits a specific attribute
Name and age are two examples for attributes	Name has to be alphabetic, age has to be positive explain the domain

The Relational Data Model and Relational Database Constraints

Relations:

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

Each attribute A_i is the name of a role played by some domain D in the relation schema R. D is called the domain of A_i and is denoted by $dom(A_i)$.

A relation schema is used to describe a relation; R is called the name of this relation. The degree (or arity) of a relation is the number of attributes n of its relation schema.

The Relational Data Model and Relational Database Constraints

Relations:

A relation of degree seven, which stores information about university students, would contain seven attributes describing each student as follows: STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)

A relation (or relation state) r of the relation schema $R(A_1, A_2, \ldots, A_n)$, also denoted by r(R), is a set of n-tuples $r = \{t_1, t_2, \ldots, t_m\}$. Each n-tuple t is an ordered list of n values $t = <v_1, v_2, ...v_n>$, where each value v_i, $1 \leq i \leq n$, is an element of $dom(A_i)$

The Relational Data Model and Relational Database Constraints

Relations:

A relation can be restated more formally using set theory concepts as follows.

A relation (or relation state) r(R) is a mathematical relation of degree n on the domains $dom(A_1)$, $dom(A_2)$, … , $dom(A_n)$, which is a subset of the Cartesian product (×) of the domains that define R:

$r(R) \subseteq (dom(A_1) \times dom(A_2) \times . . . \times (dom(A_n))$

This product of cardinalities of all domains represents the total number of possible instances or tuples that can ever exist in any relation state r(R). Of all these possible combinations, a relation state at a given time—the current relation state—reflects only the valid tuples that represent a particular state of the real world.

The Relational Data Model and Relational Database Constraints

Characteristics of Relations:

A relations implies certain characteristics that make a relation different from a file or a table. Some of these characteristics are:

1. Ordering of Tuples in a Relation:
2. Ordering of Values within a Tuple and an Alternative Definition of a Relation.
3. Values and NULLs in the Tuples
4. Interpretation (Meaning) of a Relation.

The Relational Data Model and Relational Database Constraints

Characteristics of Relations:

1. Ordering of Tuples in a Relation

- A relation is defined as a set of tuples. Mathematically, elements of a set have no order among them; hence, tuples in a relation do not have any particular order.
- Many tuple orders can be specified on the same relation. For example, tuples in the STUDENT relation could be ordered by values of Name, Ssn, Age, or some other attribute.
- The definition of a relation does not specify any order: There is no preference for one ordering over another. Hence, the relation displayed in different order is considered identical.
- When a relation is implemented as a file or displayed as a table, a particular ordering may be specified on the records of the file or the rows of the table.

The Relational Data Model and Relational Database Constraints

Characteristics of Relations:

1. Ordering of Tuples in a Relation

STUDENT

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

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

The Relational Data Model and Relational Database Constraints

Characteristics of Relations:

2. Ordering of Values within a Tuple and an Alternative Definition of a Relation:

An n-tuple is an ordered list of n values, so the ordering of values in a tuple and of attributes in a relation schema is important. However, at a more abstract level, the order of attributes and their values is not that important as long as the correspondence between attributes and values is maintained.

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)>

The Relational Data Model and Relational Database Constraints

Characteristics of Relations:

3. Values and NULLs in the Tuples:

Each value in a tuple is an atomic value; that is, it is not divisible into components within the framework of the basic relational model. Hence, composite and multivalued attributes are not allowed.

The Relational Data Model and Relational Database Constraints

Characteristics of Relations:

4. Interpretation (Meaning) of a Relation:

The relation schema can be interpreted as a declaration or a type of assertion. For example, the schema of the STUDENT relation asserts that, in general, a student entity has a Name, Ssn, Home_phone, Address, Office_phone, Age, and Gpa. Each tuple in the relation can then be interpreted as a fact or a particular instance of the assertion. For example, the first tuple asserts the fact that there is a STUDENT whose Name is Benjamin Bayer, Ssn is 305-61-2435, Age is 19, and so on.

STUDENT

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

The Relational Data Model and Relational Database Constraints

Relational Model Constraints:

Integrity constraints:

- Constrains are the restriction imposed on database.
- The various restrictions on data that can be specified on a relational database Model, Schema and Applicatons in the form of constraints is called Integrity Constraints.
- Database Integrity constraints on the database that require relations to satisfy certain properties.
- So the various integrity constraints imposed on the RDBS are categories in three.

Relational Model Constraints:

Constraints on databases can generally be divided into three main categories:

1. Inherent model-based constraints or implicit constraints - Constraints that are inherent in the data model.
2. Schema-based constraints or explicit constraints - Constraints that can be directly expressed in the schemas of the data model, typically by specifying them in the DDL (data definition language)
3. Application-based or semantic constraints or business rules - expressed and enforced by the application programs or in some other way

The Relational Data Model and Relational Database Constraints

Relational Model Constraints:

- Inherent model-based constraints or implicit constraints

The characteristics of relations are the inherent constraints of the relational model and belong to this category constraints of the relational model.

- Schema-based constraints or explicit constraints

The schema-based constraints include domain constraints, key constraints, constraints on NULLs, entity integrity constraints, and referential integrity constraints.

- Application-based or semantic constraints or business rules

Constraints that is checked within the application programs during operation of database updates i.e. Assertion, Trigger

The Relational Data Model and Relational Database Constraints

Relational Model Constraints:

- Domain constraints, key constraints, constraints on NULLs, Entity integrity constraints apply to single relations and their attributes
- Referential integrity constraints apply to multiple relations.

Another important category of constraints is data dependencies, which include functional dependencies and multivalued dependencies. They are used mainly for testing the “goodness” of the design of a relational database and are utilized in a process called normalization

The Relational Data Model and Relational Database Constraints

Relational Model Constraints:

- Domain constraints,
- Key constraints (Primary key & Foreign key),
- Constraints on NULLs
- Entity integrity constraints
- Referential integrity constraints.
- Assertion and Trigger
- Functional dependencies
- Multivalued dependencies

Domain Constraints:

- Domain constraints specify that the value of each attribute (A) must be an atomic value from the domain (A).
- The data types associated with domains i.e. numeric data types for integers, Characters, fixed length strings and variable length strings, date, time, timestamp and currency data types.
- Others possible domains may be described by a subrange of values from a data type or as an enumerated data types where all possible values are explicitly listed.

Key Constraints and Constraints on NULL Values:

- In the relational model, a relation is defined as a set of tuples. All tuples in a relation must also be distinct meaning that no two tuples can have the same combination of values for all their attributes.
- The value of a key attribute can be used to identify uniquely each tuple in the relation. A set of attributes constituting a key is a property of the relation schema; it is a constraint that should hold on every valid relation state of the schema.
- The primary key of the relation is used to identify tuples in the relation.
- Another constraint on attributes specifies whether NULL values are or are not permitted for any attributes.

Relational Databases and Relational Database Schemas:

- Domain constraints, key constraints, constraints on NULLs apply to single relations and their attributes.
- But a relational database usually contains many relations, with tuples in relations that are related in various ways. Referential integrity constraints apply to multiple relations.
- A relational database schema S is a set of relation schemas S = $\{R_1, R_2, \ldots, R_m\}$ and a set of integrity constraints IC. A relational database state DB of S is a set of relation states DB = $\{r_1, r_2, \ldots, r_m\}$ 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.

I.e. Relational database schema that we call COMPANY = {EMPLOYEE, DEPARTMENT, DEPT_LOCATIONS, PROJECT, WORKS_ON, DEPENDENT}.

Relational Databases and Relational Database Schemas:

- A relational database include both its schema and its current state. A database state that does not obey all the integrity constraints is called not valid, and a state that satisfies all the constraints in the defined set of integrity constraints IC is called a valid state.
- Integrity constraints are specified on a database schema and are expected to hold on every valid database state of that schema.
- Two other types of constraints are considered part of the relational model: entity integrity and referential integrity.

The Relational Data Model and Relational Database Constraints

Entity Integrity, Referential Integrity, and Foreign Keys:

- Entity integrity is a property that ensures that no records are duplicated and that no attributes that make up the primary key are NULL. It is one of the properties necessary to ensure the consistency of the database.
- 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.
- Key constraints and entity integrity constraints are specified on individual relations. The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations.

The Relational Data Model and Relational Database Constraints

Foreign Key constraints:

- A foreign key (FK) is an attribute or group of attributes in a database record that points to a key field or group of attributes forming a key of another database record in some (usually different) table. Usually a foreign key in one table refers to the primary key (PK) of another table.
- A foreign key constraint is a constraint that data which serves as a foreign key in one database record cannot be removed as there is still data in another record that assumes its existence.

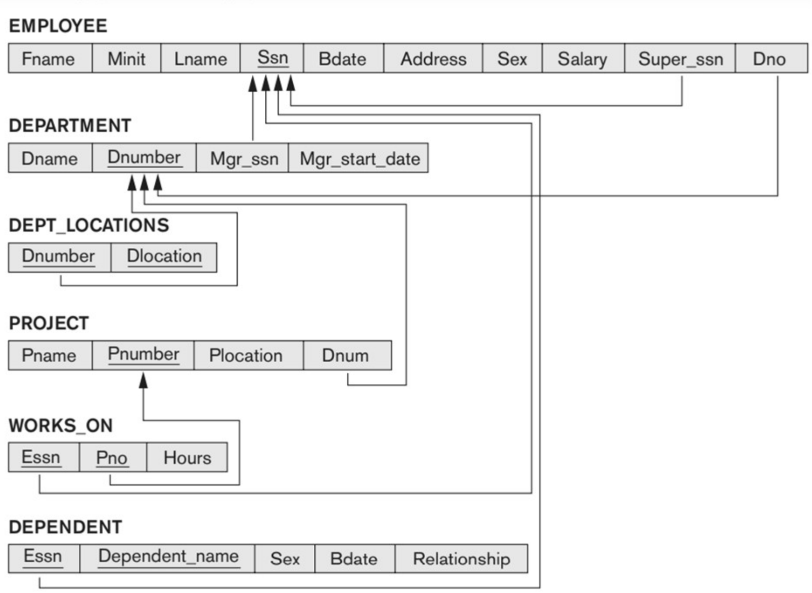
The Relational Data Model and Relational Database Constraints

Referential Integrity:

- Key constraints and entity integrity constraints are specified on individual relations. The referential integrity constraints is specified between two relations and is used to maintain the consistency among tuples of the two relations.
- For referential integrity to hold, any field in a table that is declared a foreign key can contain only values from a parent table's primary key. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity.

The Relational Data Model and Relational Database Constraints

Referential integrity constraints displayed on the COMPANY relational database schema.



The Relational Data Model and Relational Database Constraints

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

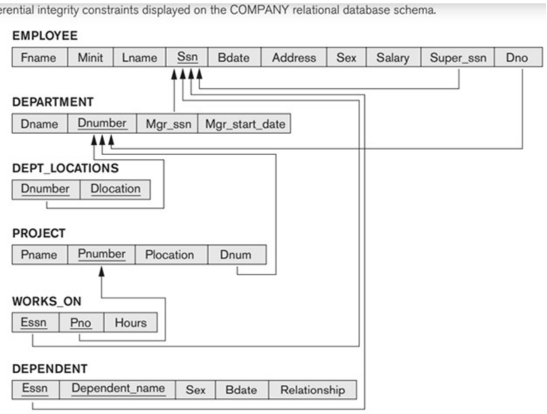
DEPENDENT

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

The Relational Data Model and Relational Database Constraints

Entity Integrity, Referential Integrity, and Foreign Keys:

Referential integrity constraints displayed on 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

Other Types of Constraints: Triggers and Assertions

An assertion is a predicate expressing a condition we wish the database to always satisfy. Domain constraints, functional dependency and referential integrity are special forms of assertion. Where a constraint cannot be expressed in these forms, we use an assertion, e.g.

- Ensuring the sum of loan amounts for each branch is less than the sum of all account balances at the branch.
- Ensuring every loan customer keeps a minimum of Rs1000 in an account.

Other Types of Constraints: Triggers and Assertions

In a DBMS, a trigger is a SQL procedure that initiates an action (i.e., fires an action) when an event (INSERT, DELETE or UPDATE) occurs.

A trigger cannot be called or executed; the DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion.

The Relational Data Model and Relational Database Constraints

Update Operations, Transactions, and Dealing with Constraint Violations;

- There are three basic database operations that can change the states of relations in the database: Insert, Delete, and Update (or Modify)
- Insert is used to insert one or more new tuples in a relation, Delete is used to delete tuples, and Update (or Modify) is used to change the values of some attributes in existing tuples.
- Whenever these operations are applied, the integrity constraints specified on the relational database schema should not be violated.
- We discuss the types of constraints that may be violated by each of these operations and the types of actions that may be taken if an operation causes a violation.

The Relational Data Model and Relational Database Constraints

The Insert Operation:

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 Relational Data Model and Relational Database Constraints

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

Notice that if a referencing attribute that causes a violation is part of the primary key, it cannot be set to NULL; otherwise, it would violate entity integrity.

The Relational Data Model and Relational Database Constraints

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 part 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.

In fact, when a referential integrity constraint is specified in the DDL, the DBMS will allow the user to choose separate options to deal with a violation caused by Delete and a violation caused by Update.

The Relational Data Model and Relational Database Constraints

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.

The Relational Data Model and Relational Database Constraints

The Transaction Concept:

- Transaction to transfer Rs 50 from account A to account B.

1. read(A)
2. A := A-50
3. write(A)
4. read(B)
5. B:= B + 50
6. write(B)

The Relational Data Model and Relational Database Constraints

Basic Relational Algebra Operations

- Unary: Select, Project, Rename;
- Binary: Set Theory(Union, Intersection, Set Difference), Cartesian Product,
- Join (Natural/Theta), Outer Join (Left/Right/Full);

The Relational Algebra and Relational Calculus

- Unary Relational Operations: SELECT, PROJECT, Rename;
- Relational Algebra Operations from Set Theory;
- Binary Relational Operations: Union, Intersection, Set Difference, Cartesian Product, DIVISION
- Join: (Natural/Theta), Outer Join (Left/Right/Full);
- Additional Relational Operations:

The Relational Algebra and Relational Calculus

- A Relational Algebra is one of procedural query language in which a user requests information from a relational database.
- Relational Algebra is a collection of operations on Relations.
- Relations are operands and the result of an operation is another relation.

Unit : The Relational Algebra and Relational Calculus

Three main types of Relational Operators:

1. **Set theory operations:**
   Union ($\cup$), **Intersection ($\cap$)**, **Difference(-)** and **Cartesian product(X)**
2. **Specific Relational Operations:**
   Selection (symbol: σ (sigma))
   Projection (symbol: π (pi))
   Join **(⋈) , Natural join(*), Outer Join (⟕) , Inner join (⋈), Full Join (⟗),**
   Division **($\div$)**
3. **Aggregate Functions**
   SUM , MINIMUM , MAXIMUM , AVERAGE, MEAN, MEDIAN , COUNT
   Aggregate functions are written using the the Script $\Im$ character as in the Elmasri/Navathe book.

Set theory operations

1. Set theory operations:

- **Union (** ∪),
- **Intersection (** ∩),
- **Difference(-)** and
- **Cartesian product(X)**

Set theory operations

Union (∪),

Consider the following relations R and S. The union operation is denoted as in set theory. It returns the union (set union) of two compatible relations. For a union operation to be legal, we require that and must have the same number of attributes. The domains of the corresponding attributes must be the same.

R

First	Last	Age
Bill	Smith	22
Sally	Green	28
Mary	Keen	23
Tony	Jones	32

S

First	Last	Age
Forrest	Gump	36
Sally	Green	28
DonJuan	DeMarco	27

Set theory operations

Union ($\cup$):

R

First	Last	Age
Bill	Smith	22
Sally	Green	28
Mary	Keen	23
Tony	Jones	32

S

First	Last	Age
Forrest	Gump	36
Sally	Green	28
DonJuan	DeMarco	27

R $\cup$ S =

First	Last	Age
Bill	Smith	22
Sally	Green	28
Mary	Keen	23
Tony	Jones	32
Forrest	Gump	36
DonJuan	DeMarco	27

Result: Relation with tuples from R and S with duplicates removed.

Set theory operations

Intersection ($\cap$):

Set intersection is denoted by ($\cap$) , and returns a relation that contains tuples that are in both of its argument relations.

R

First	Last	Age
Bill	Smith	22
Sally	Green	28
Mary	Keen	23
Tony	Jones	32

S

First	Last	Age
Forrest	Gump	36
Sally	Green	28
DonJuan	DeMarco	27

R $\cap$ S =

First	Last	Age
Sally	Green	28

Result: Relation with tuples that appear in both R and S.

Set theory operations

Difference (-):

Set difference is denoted by the minus sign (-). It finds tuples that are in one relation, but not in another. Thus results in a relation containing tuples that are in **R** but not in **S**.

Result: Relation with tuples from R but not from S

R

First	Last	Age
Bill	Smith	22
Sally	Green	28
Mary	Keen	23
Tony	Jones	32

S

First	Last	Age
Forrest	Gump	36
Sally	Green	28
DonJuan	DeMarco	27

Set theory operations

Difference (-):

Result: Relation with tuples from R but not from S

R – S =

R

First	Last	Age
Bill	Smith	22
Sally	Green	28
Mary	Keen	23
Tony	Jones	32

S

First	Last	Age
Forrest	Gump	36
Sally	Green	28
DonJuan	DeMarco	27

R - S

First	Last	Age
Bill	Smith	22
Mary	Keen	23
Tony	Jones	32

Set theory operations

Cartesian product (X):

The Cartesian product of two relations is denoted by a cross (X). The result of is a new relation with a tuple for each possible pairing of tuples. In order to avoid ambiguity, the attribute names have attached to them the name of the relation from which they came. If no ambiguity will result, we drop the relation name. The result is a very large relation. The resulting scheme is the concatenation of the schemes of and, with relation names added as mentioned.

Result: Produce all combinations of tuples from two relations.

Set theory operations

Cartesian product (X):

Result: Produce all combinations of tuples from two relations.

R

First	Last	Age
Bill	Smith	22
Mary	Keen	23
Tony	Jones	32

S

Dinner	Dessert
Steak	Ice Cream
Lobster	Cheesecake

R X S =

First	Last	Age	Dinner	Dessert
Bill	Smith	22	Steak	Ice Cream
Bill	Smith	22	Lobster	Cheesecake
Mary	Keen	23	Steak	Ice Cream
Mary	Keen	23	Lobster	Cheesecake
Tony	Jones	32	Steak	Ice Cream
Tony	Jones	32	Lobster	Cheesecake

Set theory operations

KEY POINTS TO REMEMBER TO UNION COMPATIBLE RELATIONS:

Two relations R and S are union compatible if and only if they have the same degree and the domains of the corresponding attributes are the same.

- Attributes of relations need not be identical to perform union, intersection and difference operations.
- However, they must have the same number of attributes or arity and the domains for corresponding attributes must be identical.
- Domain is the data type and size of an attribute.
- The degree of relation R is the number of attributes it contains.

Set theory operations

KEY POINTS TO REMEMBER TO UNION COMPATIBLE RELATIONS:

Some additional properties:

- Union, Intersection and difference operators may only be applied to Union Compatible relations.
- Union and Intersection are commutative operations
  $R \cup S = S \cup R$
  $R \cap S = S \cap R$
- Difference operation is NOT commutative.
  R - S not equal S - R

Relational Algebra

Three main types of Relational Operators:

1. **Set theory operations:**
   Union ($\cup$), **Intersection ($\cap$)**, **Difference(-)** and **Cartesian product(X)**
2. **Specific Relational Operations:**
   Selection (symbol: σ (sigma))
   Projection (symbol: π (pi))
   Join **($\bowtie$) , Natural join(*), Outer Join ($⟕$) , Inner join ($⟖$), Full Join ($⟗$),**
   Division **($\div$)**
3. **Aggregate Functions**
   SUM , MINIMUM , MAXIMUM , AVERAGE, MEAN, MEDIAN , COUNT
   Aggregate functions are written using the the Script $\mathfrak{I}$ character as in the Elmasri/Navathe book.

Relational Algebra

SPECIFIC RELATIONAL OPERATIONS:

Selection (σ) symbol: σ (sigma)

Projection (π) symbol: π (pi)

Join **($\bowtie$) , Natural join(*), Outer Join ($⟕$) , Inner join ($⟖$), Full Join ($⟗$),**

Division **($\div$)**

Selection

Selection (σ):

- Selection and Projection are *unary* operators.
- The selection operator is sigma: σ
- The selection operation acts like a *filter* on a relation by returning only a certain number of tuples.
- The resulting relation will have the same degree as the original relation.
- The resulting relation may have fewer tuples than the original relation.
- The tuples to be returned are dependent on a *condition* that is part of the selection operator.

Selection

Selection:
Syntax:

$\sigma_C(R)$ Returns only those tuples in R that satisfy condition **C**

A condition C can be made up of any combination of comparison or logical operators that operate on the attributes of R.

- Comparison operators: $=$ $<$ $>$ $\geq$ $\leq$ $\neq$
- Logical operators: $\wedge$ $\vee$ $\neg$

Selection

Selection:

Example:

Assume the following relation EMP has the following tuples:

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Jones	220	Econ	Adjunct
Green	160	Econ	Assistant
Brown	420	CS	Associate
Smith	500	Fin	Associate

Select only those Employees in the CS department:

$$\sigma_{\text{Dept} = \text{'CS'}} (\text{EMP})$$

Result:

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Brown	420	CS	Associate

Selection

Selection:

Example:

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Jones	220	Econ	Adjunct
Green	160	Econ	Assistant
Brown	420	CS	Associate
Smith	500	Fin	Associate

Select only those Employees who are either Assistant Professors or in the Economics department:

$$\sigma_{\text{Rank} = \text{'Assistant'} \vee \text{Dept} = \text{'Econ'}} (\text{EMP})$$

Result:

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Jones	220	Econ	Adjunct
Green	160	Econ	Assistant

Projection

Projection (π)

- Projection is also a Unary operator.
- The Projection operator is pi: π
- Projection limits the *attributes* that will be returned from the original relation.
- The general syntax is: $\pi_{\text{attributes}}$ (R)
  Where *attributes* is the list of attributes to be displayed and R is the relation.
- The resulting relation will have the same number of tuples as the original relation (unless there are duplicate tuples produced).
- The degree of the resulting relation may be equal to or less than that of the original relation.

Projection

Projection: Example

EMP

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Jones	220	Econ	Adjunct
Green	160	Econ	Assistant
Brown	420	CS	Associate
Smith	500	Fin	Associate

Project only the names and departments of the employees:

$\pi_{\text{name, dept}}$ (EMP)

Name	Dept
Smith	CS
Jones	Econ
Green	Econ
Brown	CS
Smith	Fin

Selection and Projection

Combining Selection and Projection:

Example: EMP

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Jones	220	Econ	Adjunct
Green	160	Econ	Assistant
Brown	420	CS	Associate
Smith	500	Fin	Associate

The selection and projection operators can be combined to perform both operations.

Show the names of all employees working in the CS department:

$\pi_{name}(\sigma_{Dept = 'CS'}(EMP))$

Results:

Name
Smith
Brown

ASSIGNMENT operation and RENAME Operation (← (left arrow))

To retrieve the first name, last name, and salary of all employees who work in department number 5, we must apply a SELECT and a PROJECT operation. We can write a single relational algebra expression, also known as an in-line expression, as follows:

$$\pi_{Fname,\ Lname,\ Salary}(\sigma_{Dno=5}(EMPLOYEE))$$

Alternatively, we can explicitly show the sequence of operations, giving a name to each intermediate relation, and using the assignment operation, denoted by ← (left arrow), as follows:

$$DEP5\_EMPS \leftarrow \sigma_{Dno=5}(EMPLOYEE)$$

$$RESULT \leftarrow \pi_{Fname,\ Lname,\ Salary}(DEP5\_EMPS)$$

ASSIGNMENT operation and RENAME Operation (← (left arrow))

We can also use this technique to rename the attributes in the intermediate and result relations. This can be useful in connection with more complex operations such as UNION and JOIN. To rename the attributes in a relation, we simply list the new attribute names in parentheses, as in the following example:

TEMP ← σDno=5(EMPLOYEE)

R(First_name, Last_name, Salary) ← πFname, Lname, Salary(TEMP)

We can also define a formal RENAME operation - which can rename either the relation name or the attribute names, or both—as a unary operator.

PROJ_DEPT ← PROJECT * ρ(Dname, Dnum, Mgr_ssn, Mgr_start_date)(DEPARTMENT)

DEPT ← ρ(Dname, Dnum, Mgr_ssn, Mgr_start_date)(DEPARTMENT)

Join

Join:

Syntax:

- The generic join operator (called the *Theta Join* is: $\bowtie$)
- It takes as arguments the attributes from the two relations that are to be joined.
- For example assume we have the EMP relation as above and a separate DEPART relation with (Dept, MainOffice, Phone) :
- EMP $\underset{\text{Condition}}{\bowtie}$ DEPART

  E.g. EMP $\underset{\text{EMP.Dept = DEPART.Dept}}{\bowtie}$ DEPART

- The join condition can be $=$ $<$ $>$ $\geq$ $\leq$ $\neq$
- When the join condition operator is $=$ then we call this an *Equijoin*

Join

Join:Example

Assume we have the EMP relation from above and the following DEPART relation:

EMP

Name	Office	Dept	Rank
Smith	400	CS	Assistant
Jones	220	Econ	Adjunct
Green	160	Econ	Assistant
Brown	420	CS	Associate
Smith	500	Fin	Associate

DEPART

Dept	MainOffice	Phone
CS	404	555-1212
Econ	200	555-1234
Fin	501	555-4321
Hist	100	555-9876

Find all information on every employee including their department info:

$$\text{EMP} \bowtie_{\text{EMP.Dept = DEPART.Dept}} \text{DEPART}$$

Results:

Name	Office	EMP.Dept	Rank	DEPART.Dept	MainOffice	Phone
Smith	400	CS	Assistant	CS	404	555-1212
Jones	220	Econ	Adjunct	Econ	200	555-1234
Green	160	Econ	Assistant	Econ	200	555-1234
Brown	420	CS	Associate	CS	404	555-1212
Smith	500	Fin	Associate	Fin	501	555-4321

Natural Join

Natural Join (*)

- Notice in the generic (Theta) join operation, any attributes in common (such as dept above) are repeated.
- The *Natural Join* operation removes these duplicate attributes.
- The natural join operator is: *
- We can also assume using * that the join condition will be **=** on the two attributes in common.

Example: EMP * DEPART
Results:

Name	Office	Dept	Rank	MainOffice	Phone
Smith	400	CS	Assistant	404	555-1212
Jones	220	Econ	Adjunct	200	555-1234
Green	160	Econ	Assistant	200	555-1234
Brown	420	CS	Associate	404	555-1212
Smith	500	Fin	Associate	501	555-4321

Join

Outer Join:

- In the Join operations so far, only those tuples from both relations that satisfy the join condition are included in the output relation.
- The *Outer join* includes other tuples as well according to a few rules.
- Three types of outer joins:
  1. **Left Outer Join** ⟕ includes all tuples in the left hand relation and includes only those matching tuples from the right hand relation.
  2. **Right Outer Join** ⟖ includes all tuples in the right hand relation and includes ony those matching tuples from the left hand relation.
  3. **Full Outer Join** ⟗ includes all tuples in the left hand relation and from the right hand relation.

Join

Examples: Assume we have two relations: PEOPLE and MENU

PEOPLE:

Name	**Age**	**Food**
Alice	21	Hamburger
Bill	24	Pizza
Carl	23	Beer
Dina	19	Shrimp

MENU:

Food	**Day**
Pizza	Monday
Hamburger	Tuesday
Chicken	Wednesday
Pasta	Thursday
Tacos	Friday

PEOPLE ⟕ $_{\text{people.food = menu.food}}$ MENU

Name	**Age**	**people.Food**	**menu.Food**	**Day**
Alice	21	Hamburger	Hamburger	Tuesday
Bill	24	Pizza	Pizza	Monday
Carl	23	Beer	*NULL*	*NULL*
Dina	19	Shrimp	*NULL*	*NULL*

Join

Examples: Assume we have two relations: PEOPLE and MENU

PEOPLE:

Name	Age	Food
Alice	21	Hamburger
Bill	24	Pizza
Carl	23	Beer
Dina	19	Shrimp

MENU:

Food	Day
Pizza	Monday
Hamburger	Tuesday
Chicken	Wednesday
Pasta	Thursday
Tacos	Friday

- PEOPLE $⟖_{people.food\ =\ menu.food}$ MENU

Name	Age	people.Food	menu.Food	Day
Bill	24	Pizza	Pizza	Monday
Alice	21	Hamburger	Hamburger	Tuesday
NULL	*NULL*	*NULL*	Chicken	Wednesday
NULL	*NULL*	*NULL*	Pasta	Thursday
NULL	*NULL*	*NULL*	Tacos	Friday

Join

Examples: Assume we have two relations: PEOPLE and MENU

PEOPLE:

Name	Age	Food
Alice	21	Hamburger
Bill	24	Pizza
Carl	23	Beer
Dina	19	Shrimp

MENU:

Food	Day
Pizza	Monday
Hamburger	Tuesday
Chicken	Wednesday
Pasta	Thursday
Tacos	Friday

- PEOPLE $⟗_{people.food\ =\ menu.food}$ MENU

Name	Age	people.Food	menu.Food	Day
Alice	21	Hamburger	Hamburger	Tuesday
Bill	24	Pizza	Pizza	Monday
Carl	23	Beer	*NULL*	*NULL*
Dina	19	Shrimp	*NULL*	*NULL*
NULL	*NULL*	*NULL*	Chicken	Wednesday
NULL	*NULL*	*NULL*	Pasta	Thursday
NULL	*NULL*	*NULL*	Tacos	Friday

Division (÷)

Examples: See the tables Completed, DBProject and their

division: Completed ÷ DBProject

The division is a binary operation that is written as $R \div S$. Division is not implemented directly in SQL.

If *DBProject* contains all the tasks of the Database project, then the result of the division above contains exactly the students who have completed both of the tasks in the Database project.

Completed

Student	Task
Fred	Database1
Fred	Database2
Fred	Compiler1
Eugene	Database1
Eugene	Compiler1
Sarah	Database1
Sarah	Database2

DBProject

Task
Database1
Database2

Completed ÷ DBProject

Student
Fred
Sarah

Division (÷)

Examples:

Retrieve the names of employees who work on all the projects that 'John Smith' works on.

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

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

(a)

SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

Pno
1
2

SSNS

Ssn
123456789
453453453

(b)

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

Figure 6.8
The DIVISION operation. (a) Dividing SSN_PNOS by SMITH_PNOS. (b) $T \leftarrow R \div S$.

$\text{SMITH} \leftarrow \sigma_{\text{Fname='John' AND Lname='Smith'}}(\text{EMPLOYEE})$
$\text{SMITH\_PNOS} \leftarrow \pi_{\text{Pno}}(\text{WORKS\_ON} \bowtie_{\text{Essn=Ssn}} \text{SMITH})$

$\text{SSN\_PNOS} \leftarrow \pi_{\text{Essn, Pno}}(\text{WORKS\_ON})$

$\text{SSNS(Ssn)} \leftarrow \text{SSN\_PNOS} \div \text{SMITH\_PNOS}$
$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname}}(\text{SSNS} * \text{EMPLOYEE})$

Aggregate Function Examples

Aggregate Functions and Grouping:

We can define an AGGREGATE FUNCTION operation, using the symbol $\mathfrak{I}$ (pronounced script F). The syntax is as follows:

$$\langle\text{grouping attributes}\rangle \; \mathfrak{I} \; \langle\text{function list}\rangle \; (R)$$

where <grouping attributes> is a list of attributes of the relation specified in R, and <function list> is a list of (<function> <attribute>) pairs. In each such pair, <function> is one of the allowed functions—such as SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT—and <attribute> is an attribute of the relation specified by R. The resulting relation has the grouping attributes plus one attribute for each element in the function list. Examples:

The aggregate function operation.

a. $\rho_{R(\text{Dno, No\_of\_employees, Average\_sal})}({}_{\text{Dno}}\,\mathfrak{I}_{\text{COUNT Ssn, AVERAGE Salary}}\,(\text{EMPLOYEE}))$.

b. ${}_{\text{Dno}}\,\mathfrak{I}_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

c. $\mathfrak{I}_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

Aggregate Function Examples

Aggregate Functions and Grouping:

$$\langle\text{grouping attributes}\rangle \; \mathfrak{I} \; \langle\text{function list}\rangle \; (R)$$

The aggregate function operation.

a. $\rho_{R(\text{Dno, No\_of\_employees, Average\_sal})}({}_{\text{Dno}}\,\mathfrak{I}_{\text{COUNT Ssn, AVERAGE Salary}}\,(\text{EMPLOYEE}))$.

b. ${}_{\text{Dno}}\,\mathfrak{I}_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

c. $\mathfrak{I}_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

R

(a)

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

(b)

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

(c)

Count_ssn	Average_salary
8	35125

Aggregate Function Examples

Aggregate Functions and Grouping:

$$\langle \text{grouping attributes} \rangle\ \Im\ \langle \text{function list} \rangle\ (R)$$

$$\text{Dno}\ \Im\ _{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$$

If no grouping attributes are specified, the functions are applied to all the tuples in the relation, so the resulting relation has a single tuple only. For example

$$\Im\ _{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$$

Aggregate Function Examples

Assume the relation EMP has the following tuples:

Name	**Office**	**Dept**	**Salary**
Smith	400	CS	45000
Jones	220	Econ	35000
Green	160	Econ	50000
Brown	420	CS	65000
Smith	500	Fin	60000

Q1.Find the minimum Salary: $\Im_{\text{MIN (salary)}}(\text{EMP})$
Results:

MIN(salary)
35000

Q2.Find the average Salary: $\Im_{\text{AVG (salary)}}(\text{EMP})$
Results:

AVG(salary)
51000

Q3.Count the number of employees in the CS department: $\Im_{\text{COUNT (name)}}(\sigma_{\text{Dept = 'CS'}}(\text{EMP}))$
Results:

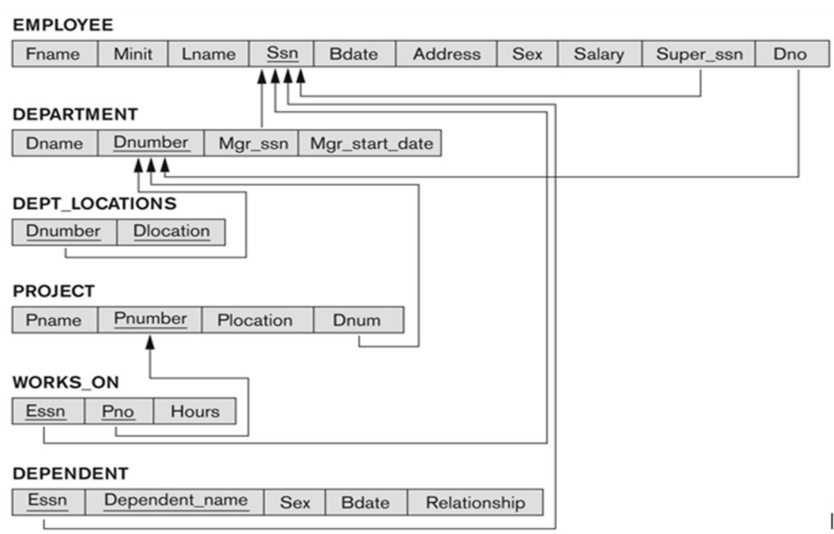
COUNT(name)
2

Q4.Find the total payroll for the Economics department: $\Im_{\text{SUM (salary)}}(\sigma_{\text{Dept = 'Econ'}}(\text{EMP}))$
Results:

SUM(salary)
85000

Operations of Relational Algebra

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}),(\text{<join attributes 2>})} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}),(\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$



The Relational Data Model and Relational Database Constraints

Basic Relational Algebra Operations

Exercises:

The Relational Data Model and Relational Database Constraints

Exercise: Retrieve the names of employees who work on all the projects that 'John Smith' works on.

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

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

1. Write the result relations SMITH and SMITH_PNOS of following relational algebra query of
   $\text{SMITH} \leftarrow \sigma_{\text{Fname='John' AND Lname='Smith'}}(\text{EMPLOYEE})$
   $\text{SMITH\_PNOS} \leftarrow \pi_{\text{Pno}}(\text{WORKS\_ON} \bowtie_{\text{Essn=Ssn}} \text{SMITH})$

2. Write the result relations SSN_PNOS of following relational algebra query of $\text{SSN\_PNOS} \leftarrow \pi_{\text{Essn, Pno}}(\text{WORKS\_ON})$

3. Write the result relations SSNS(Ssn) and RESULT of following relational algebra query of
   $\text{SSNS(Ssn)} \leftarrow \text{SSN\_PNOS} \div \text{SMITH\_PNOS}$
   $\text{RESULT} \leftarrow \pi_{\text{Fname, Lname}}(\text{SSNS} * \text{EMPLOYEE})$

The Relational Data Model and Relational Database Constraints

Exercise: Select the EMPLOYEE tuples whose department is 10, or those whose salary is greater than Rs50,000.

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

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

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

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

The Relational Data Model and Relational Database Constraints

Exercise: Select the tuples for all employees who either work in department 4 and salary over Rs25,000, or work in department 5 and salary over Rs 30,000.

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

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

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

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

$\sigma_{(Dno=4 \text{ AND } Salary>25000) \text{ OR } (Dno=5 \text{ AND } Salary>30000)}(EMPLOYEE)$

The Relational Data Model and Relational Database Constraints

Exercise: Retrieve the first name, last name, and salary of all employees who work in department number 5.

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

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

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

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

The Relational Data Model and Relational Database Constraints

Exercise: Retrieve the name (Fname, Lname) of the manager of each department.

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

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

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

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

The Relational Data Model and Relational Database Constraints

Exercise: Retrieve the Social Security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5.

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

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

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

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

The Relational Data Model and Relational Database Constraints

Exercise: Retrieve the Social Security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5.

- Retrieve the Ssn of employees who is working in department no 5.
- Retrieve the supervisor Ssn (Super_ssn) of employees
- UNION of both.

DEP5_EMPS ← σDno=5(EMPLOYEE)

RESULT1 ← πSsn(DEP5_EMPS)

RESULT2(Ssn) ← πSuper_ssn(DEP5_EMPS)

RESULT ← RESULT1 ∪ RESULT2

Exercise: List the names of managers who have at least one dependent.

MGRS(Ssn) ← πMgr_ssn(DEPARTMENT)

EMPS_WITH_DEPS(Ssn) ← πEssn(DEPENDENT)

MGRS_WITH_DEPS ← (MGRS $\cap$ EMPS_WITH_DEPS)

RESULT ← πLname, Fname(MGRS_WITH_DEPS * EMPLOYEE)

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

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

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