Information Management II

3. Database Models

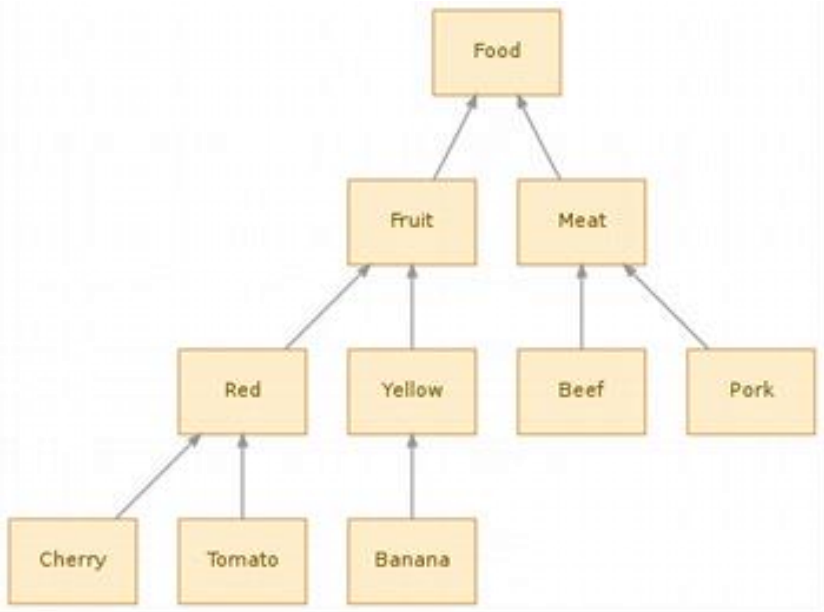
Yvette Graham
Yvette.graham@tcd.ie

DBMS Classification

- Main method of DBMS classification is via the conceptual data model used
- The choice of model affects virtually all other components in the system
  - Particularly the external schemas and associated DML
- Examples
  - Hierarchical
  - Network
  - Relational
  - Object-oriented and Object-Relational
  - Graph, Columnar, In Memory, NoSQL….

Hierarchical Database

- One of the oldest database models
  - Commonly used in Mainframe computing
- Organised hierarchically with parent and child nodes (like a family tree!)



Network Database

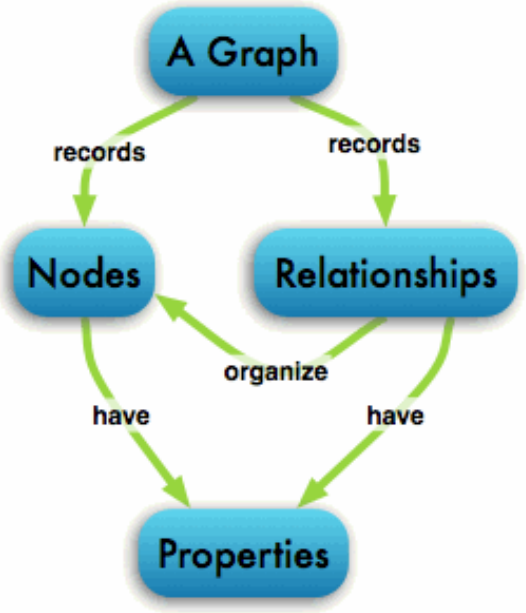
- Also have a hierarchical structure
- Uses “members” and “owners” rather than “parents” and “children”.
- Each member can have more than one owner

Object-Oriented Database

- Attempts to Model Data Storage in a similar fashion to application programs
  - Persistent storage of program objects such as class definitions
  - Objects can survive past the end of program execution
- Impedance Mismatch Problem
  - Data Structures in DBMS incompatible with the programming language's Data Structures

Graph Database

- Uses a graph structure with:
  - Nodes
  - Edges
  - Properties



- Graph databases treat the relationship between things as equally important to the things themselves.

Relational Database

- Differs from previous models as it is not Hierarchical, but Relational
- More flexible than either the hierarchical or network database models.
- Uses notions of:
  - Relations (Tables)
  - Tuples (Rows)
  - Attributes (Columns)

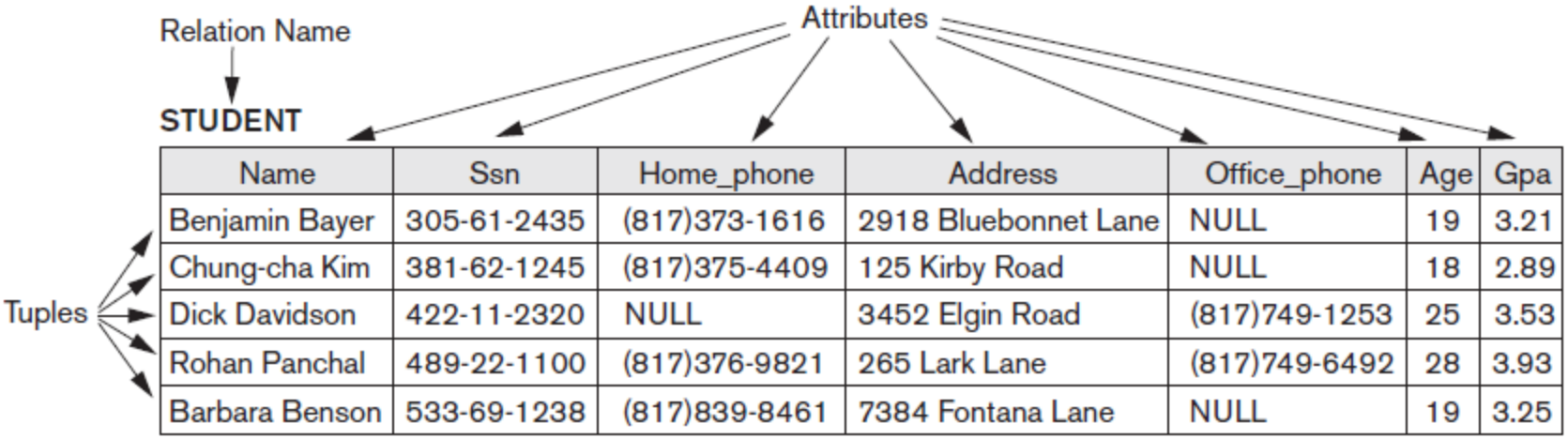
Relational Databases

- The Relational Model
  - First Introduced in 1969
  - Theoretical Basis
    - Set Theory
    - First-Order Predicate Logic
- Database represented as a collection of *mathematical relations*
  - Informally, relations resemble tables of values

The Relational Model

- The *table*, or *relation*, is the basic storage structure of a Relational Database.
  - Tables are “Two-Dimensional”
- Each *row*, or *tuple*, in a table represents a collection of related values
  - A row represents a fact that corresponds to an entity or relationship in the real world
- Each *column*, or *attribute*, contains values of the same data type

The Relational Model



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 Model

- Domain
  - The data type describing the values that can appear in each column is represented by a *domain* of possible values
  - mobile_phone_number: The set of 10 digit phone numbers valid in Ireland
  - PPS_number: 9 characters in length. 7 numeric characters in positions 1 to 7, followed by 1 alphabetic check character in position 8, and either a space or the letter “W” in position 9

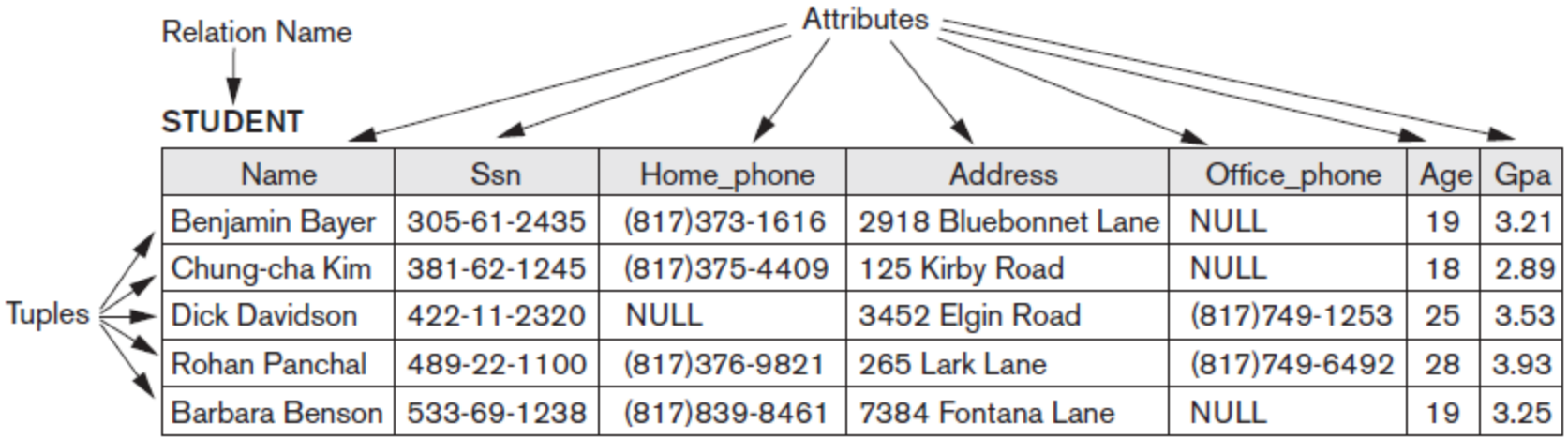
EMPNO	NAME	JOB	DEPTNO
7856	MCNULTY	OFFICER	30
7710	DANIELS	LIEUTENANT	40
7992	GREGGS	DETECTIVE	10
7428	MORELAND	DETECTIVE	20

DEPTNO	NAME	LOCATION
10	NARCOTICS	TOWER 221
20	HOMICIDE	CITY CENTRE
30	MARINE	DOCKS
40	EVIDENCE	DOWNTOWN

Formal Definition

- A relational schema R, denoted by $R(A_1, A_2, \ldots, A_n)$ is made up of:
  - *relation* name R
  - List of *attributes* $A_1 \ldots A_n$
  - Each *attribute* A_i is the name of the role played by *domain* D_i in the *relation* R
    - D_i is the *domain* of A_i and is denoted by $dom(A_i)$
  - The *degree* of a schema, is equal to the number of *attributes*, n

Formal Definiton



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)
- The *degree* of the relation STUDENT is….
- $\mathrm{dom}(\mathrm{Ssn})$ = ….

Formal Definition

- A *relation state* r of a relational schema $R(A_1, A_2, \ldots, A_n)$ also denoted $r(R)$ is:
  - *A set of tuples* $r = <t_1, t_2, \ldots, t_m>$
  - Each *tuple* t is an ordered list of n values $t = <v_1, v_2, \ldots, v_n>$
    - where each value v_i, $1 \leq i \leq n$, is an element of $dom(A_i)$
  - The i^{th} value of tuple t_n, which corresponds to attribute A^i, is referred to as $t_n[A_i]$ or $t_n[i]$

Formal Definition

$t_3 = <$Dick Davidson,422-11-2320,NULL,3452 Elgin Road,(817)749-1253,25,3.53$>$

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

$t_5[A_3] = (817)839\text{-}8461$

relation state = $r(R) = <t_1, t_2, t_3, t_4, t_5>$

Characteristics of Relations

- Ordering of tuples in a relation
  - A Relation defined as a *set* of tuples
  - Elements of a *set* have no order among them

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

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

Quick Task

- Suggest a relational table for a company wishing to manage its sales persons and customer records.

Suppose that in the database:

- The database must contain the following information: customer numbers (Ids) , salesperson numbers (ids), customer names and salesperson names
- For Each customer, the database stores his/her name and the sales person who services that customer
- For each customer there is only one sales person

Characteristics of Relations

- Ordering of tuples in a relation
  - A Relation defined as a *set* of tuples
  - Elements of a *set* have no order among them
- Ordering of values within a tuple
  - Each *tuple* t is an ordered list of n values
    $t = <v_1, v_2, \ldots, v_n>$
  - Order can change as long as correspondence between attributes and values is maintained

Characteristics of Relations

- Values in tuples
  - Each value in a tuple is *atomic*
    - For example: Student Age
    - Composite and multivalued attributes not allowed in the "Flat" Relational Model
  - Multivalued attributes
    - For example: College Degree
    - Must be represented by separate relations
  - Composite attributes
    - For example: Address
    - Represented only by simple component attributes in basic relational model

Characteristics of Relations

- NULL values
  - Represent the values of attributes that may be unknown or may not apply to a tuple
  - Meanings for NULL values
    - Value unknown
    - Value exists but is not available
    - Attribute does not apply to this tuple (also known as value undefined)
  - The NULL value is defined for each domain and there are restrictions

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 Constraints

- Restrictions on the actual values that can be placed in a database state
- These rules are derived from the rules of the world that the database represents
- Constraints inherent in the data model
  - *Inherent model-based* or *implicit constraints*

Relational Model Constraints

- Constraints expressed in the schemas of the data model i.e. DDL
  - *Schema-based* or *explicit constraints*
- Constraints that cannot be expressed in the DDL
  - Must be enforced by the application programs
  - *Application-based* or *semantic constraints*, *Business Rules*

Keys and Integrity Constraints

- A Relational DB consists of many relations
  - tuples of those relations can be related in various ways
- Every relation and every attribute has a name
  - As a result, can be uniquely identified
- Attribute names are often qualified by relation name
  - Resolves ambiguity
    - PATIENT.name
    - DOCTOR.name

Primary Key

- Most relations have one attribute whose values uniquely identify its tuples
  - e.g. student_number in the relation STUDENT
  - no two students can have the same student number
- This attribute is known as a *key*
  - More specifically, this type of *key* is called a *Primary Key*

Primary Key

- Not every relation uses a single attribute as its Primary Key

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

- When multiple Candidate Keys exist, they may be combined, or one chosen, to form a Primary Key

Entity Integrity Constraint

- Specifies that there may not be any duplicate entries in the Primary Key attribute
- NULL values are not permitted in Primary Key fields
  - Primary Key is used to identify a tuple
  - Having a NULL in a Primary Key implies that we cannot identify some tuples
- Once defined, Key and Entity Constraints are enforced by the DBMS

Referential Integrity

- Key and Entity Constraints are specified on individual relations
- Referential Integrity Constraints are specified between two relations
  - Maintains consistency among tuples in the two relations
- Informally:
  - A tuple in one relation that refers to another relation, must refer to an *existing tuple* in that relation

Referential Integrity

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

Foreign Keys

- A Foreign Key formally specifies a Referential Integrity Constraint between two relations
- Consider two relation schemas R_1 and R_2
- A set of attributes FK in R_1 is a Foreign Key of R_1 that references R_2 if:
  - The attributes of FK have the same domains as the Primary Key attributes PK of R_2
    - FK is said to reference or refer to R_2
  - A value of FK in a tuple t1 either occurs as a value of PK for some tuple t2, or is NULL
    - tuple t_1 is said to reference or refer to tuple t_2

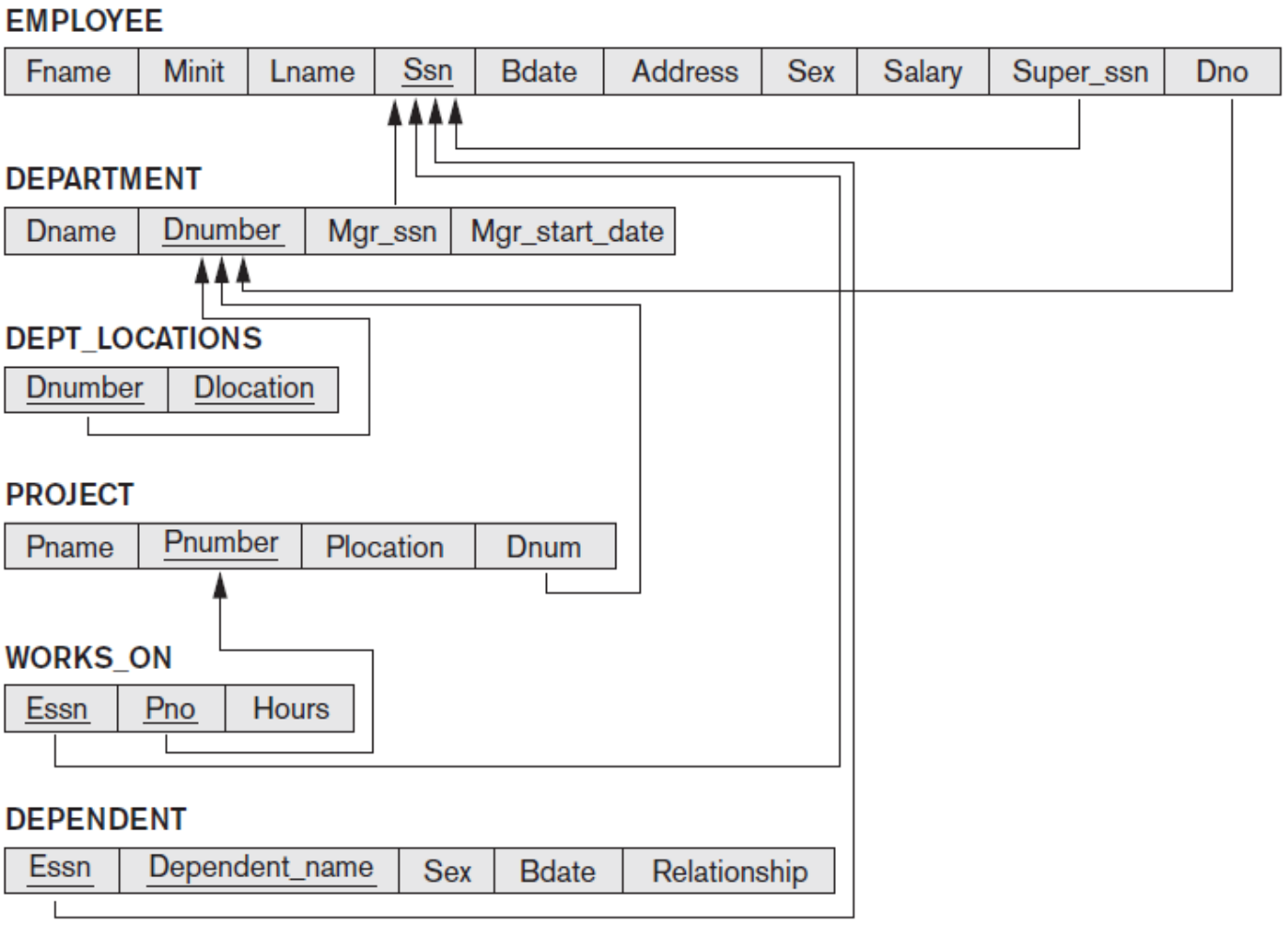
Table Relationships

EMPNO	NAME	JOB	DEPTNO
7856	MCNULTY	OFFICER	30
7710	DANIELS	LIEUTENANT	40
7992	GREGGS	DETECTIVE	10
7428	MORELAND	DETECTIVE	20

Foreign Key

Primary Key

DEPTNO	NAME	LOCATION
10	NARCOTICS	TOWER 221
20	HOMICIDE	CITY CENTER
30	MARINE	DOCKS
40	EVIDENCE	DOWNTOWN


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

Example

STUDENT

(student_number, student_name, student_address)

COURSE

(course_number, course_title, lecturer)

RESULT

(course_number, student_number, grade)

Example

STUDENT

(student_number, student_name, student_address)

COURSE

(course_number, course_title, lecturer)

RESULT

(course_number, student_number, grade)

Example

- fk_course_number is a FK of RESULT that references COURSE
  - RESULT.course_number and COURSE.course_number have the same domain
  - Each tuple in RESULT must contain a course_number that exists in a tuple in COURSE, or be NULL

Example

- fk_student_number is a FK of RESULT that references STUDENT
  - RESULT.student_number and STUDENT.student_number have the same domain
  - Each tuple in RESULT must contain a student_number that exists in a tuple in STUDENT, or be NULL

Example

STUDENT

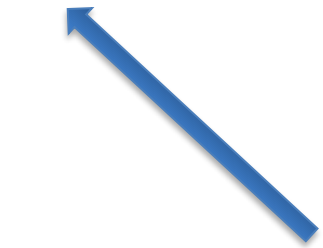
(student_number, student_name, student_address)

RESULT

(course_number, student_number, grade)

COURSE

(course_number, course_title, lecturer)



EMPNO	NAME	JOB	MGR	HIREDATE	SALARY	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10