Relational Database Model and Constraints

Flat Database

- Issues ?
  - Difficult to search
  - Difficult to generate reports
  - Difficult to sort data based on particular field

Informal Terms		Formal Terms
Table		Relation
Column		Attribute/Domain
Row		Tuple
Values in a column		Domain
Table Definition		Schema of a Relation
Populated Table		Extension

Relational Database

- Relational databases use **tables** to store information.

LName	FName	City	Age	Salary
Smith	John	New York	35	$280
Doe	Jane	Boston	28	$325
Brown	Scott	New York	41	$265
Howard	Shemp	Los Angeles	48	$359
Taylor	Tom	London	22	$250

- *Issue with this table ?*

- **Primary Key:** uniquely identifies each record in the table

Relational Database

- “Relational” part ->
  - how multiple tables relate to each other,
  - allowing a user to extract information from a combination of tables

LName	FName	City	Age	Salary
Smith	John	3	35	$280
Doe	Jane	1	28	$325
Brown	Scott	3	41	$265
Howard	Shemp	4	48	$359
Taylor	Tom	2	22	$250

City #	City Name
1	Boston
2	London
3	New York
4	Los Angeles

Relational Database vs Spreadsheet

- Relational database is a **big spreadsheet**
  - that several people can update simultaneously
- Each *table* in the database is one spreadsheet

- **Difference:**
  - RD is more restrictive than a spreadsheet
    - All the data in one column must be of the same type, e.g., integer, decimal, character string, or date.
  - Rows in an RD are not ordered
  - RD allow user to enforce constraints

- Example: Mailing List

Formal Definitions

- **Schema** of a Relation: *R* (A1, A2, ...An)
    - *R* is defined over **attributes** A1, A2, ...An

- Example:

CUSTOMER (C-id, C-name, Address, Phone#)

- Each attribute has a **domain** or a set of valid values.
    - Example: domain of C-id is 6 digit numbers.

Formal Definitions

- Relation is formed over the Cartesian product of the sets
- Each set has values from a domain
- Domain is used in a specific role which is conveyed by the attribute name.

- Example:
  - C-name is defined over the domain of strings of 25 characters.
  - Role these strings play: name of customers.

Formal Definitions

R(A1, A2, ..., An)

r(R) ⊂ dom (A1) X dom (A2) X ....X dom(An)

- R: schema of the relation
- r of R: a specific "value" or population of R.
- R is called the **intension** of a relation
- r is called the **extension** of a relation

- Example:
  - Let S1 = {0,1} and S2 = {a,b,c}
  - Let R ⊂ S1 X S2
  - r(R) = {<0,a> , <0,b> , <1,c> } is one possible “state”

Characteristics Of Relations

- Ordering of tuples in a relation r(R):
  - tuples are *not considered to be ordered*
- Ordering of attributes in a relation schema R
  - attributes in R(A1, A2, ..., An) and the values in t=<v1, v2, ..., vn> are considered to be ordered .

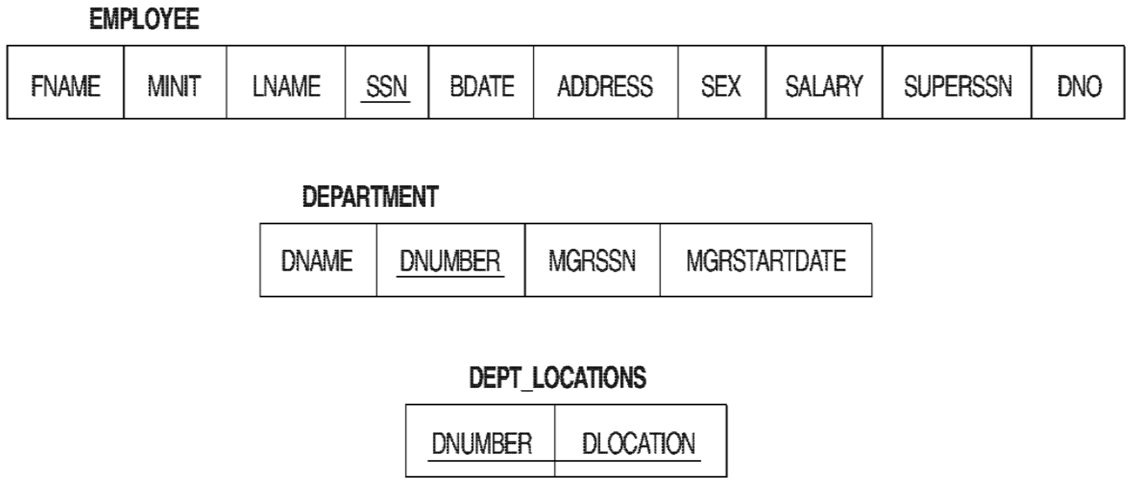
Characteristics Of Relations

- Values in a tuple:
  - Atomic (indivisible).
  - Null value

- Domains of all attributes should be indivisible
  - Integers and strings,
  - Record structures: set, list, array … ?

Why we need Multiple Tables

- Why not put all attributes in one relation ?



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

- Dup
- Primary key
- Cannot insert value for employee with no assigned department … Null value in PK

Constraints

Relational Database Constraints

- Generally, there are many constraints on the values in a database state.
- Constraints are derived from the rules in the mini-world

Up-till now we have seen constraints

- **No duplicate tuples** (rows)
- **Unique id** (primary key)
- **Atomic value** for domain of an attribute

Categories of Constraints

- **Model-based constraints or implicit constraints.**
  - These are inherent in data model, like no duplicate rows in RDBMs table.
- **Schema-based or explicit constraints.**
  - Can be expressed directly in the schema using DDL
- **Application based or semantic constraints or business rules.**
  - Can’t be expressed directly in the schema
  - Must be enforced by the application programs

Schema-Based constraints

- **Key** constraints
- **Entity integrity** constraints
- **Referential integrity** constraints

Key Constraint

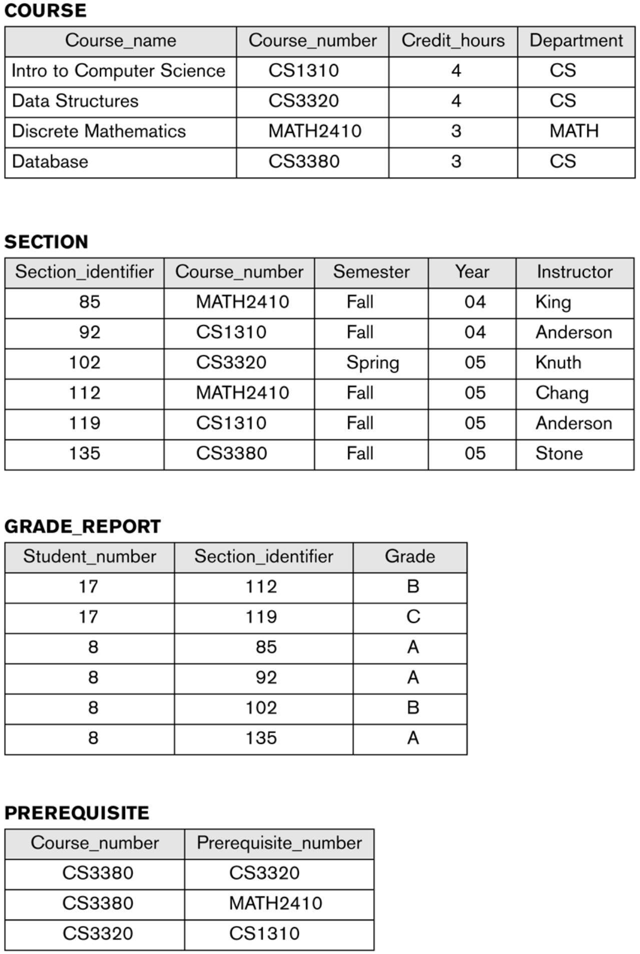
- Relation is a **set** of tuples. Thus, all tuples must be distinct as all elements in the set are distinct

- **Example 1:** Student Relation

Student(Name, SSN, Address, Phone, Major, Age, Gpa)

  - **KEY is** SSN, as it uniquely identify the student.
  - Consider set of attributes {Name, SSN, Age},
    - This set uniquely identify the student.
    - This is **SUPERKEY**

Example of a university database



COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Key Constraint

- **Superkey**:
  - For any distinct tuples t1 & t2 in r(R),

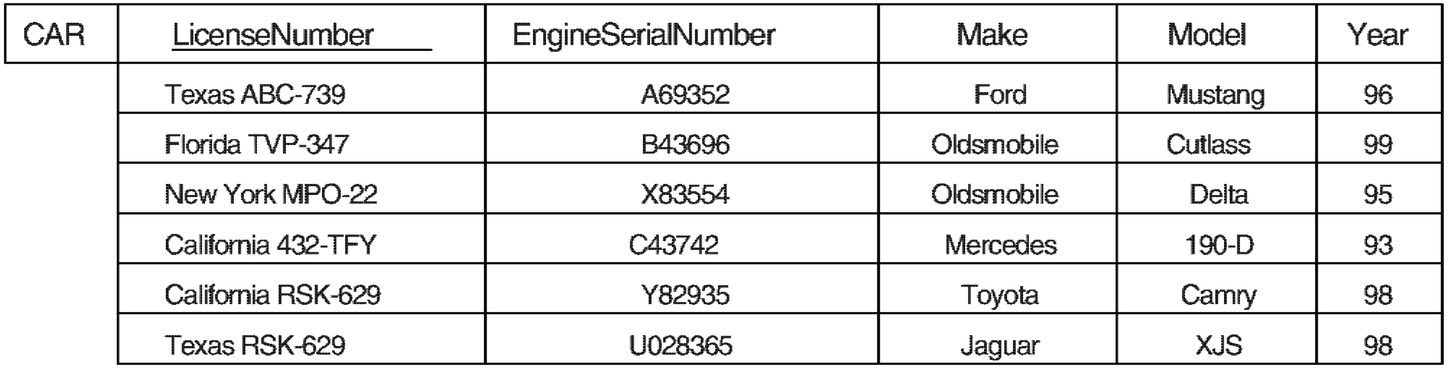
    $$t1[SK] \neq t2[SK].$$
- **Key**:
  - A "minimal" superkey;
  - Key is a superkey from which we cannot remove any attribute and still have the uniqueness constraint hold.

- **Example 2: CAR relation schema**

CAR(LicenseNo, Engine_SerialNo, Make, Model, Year)

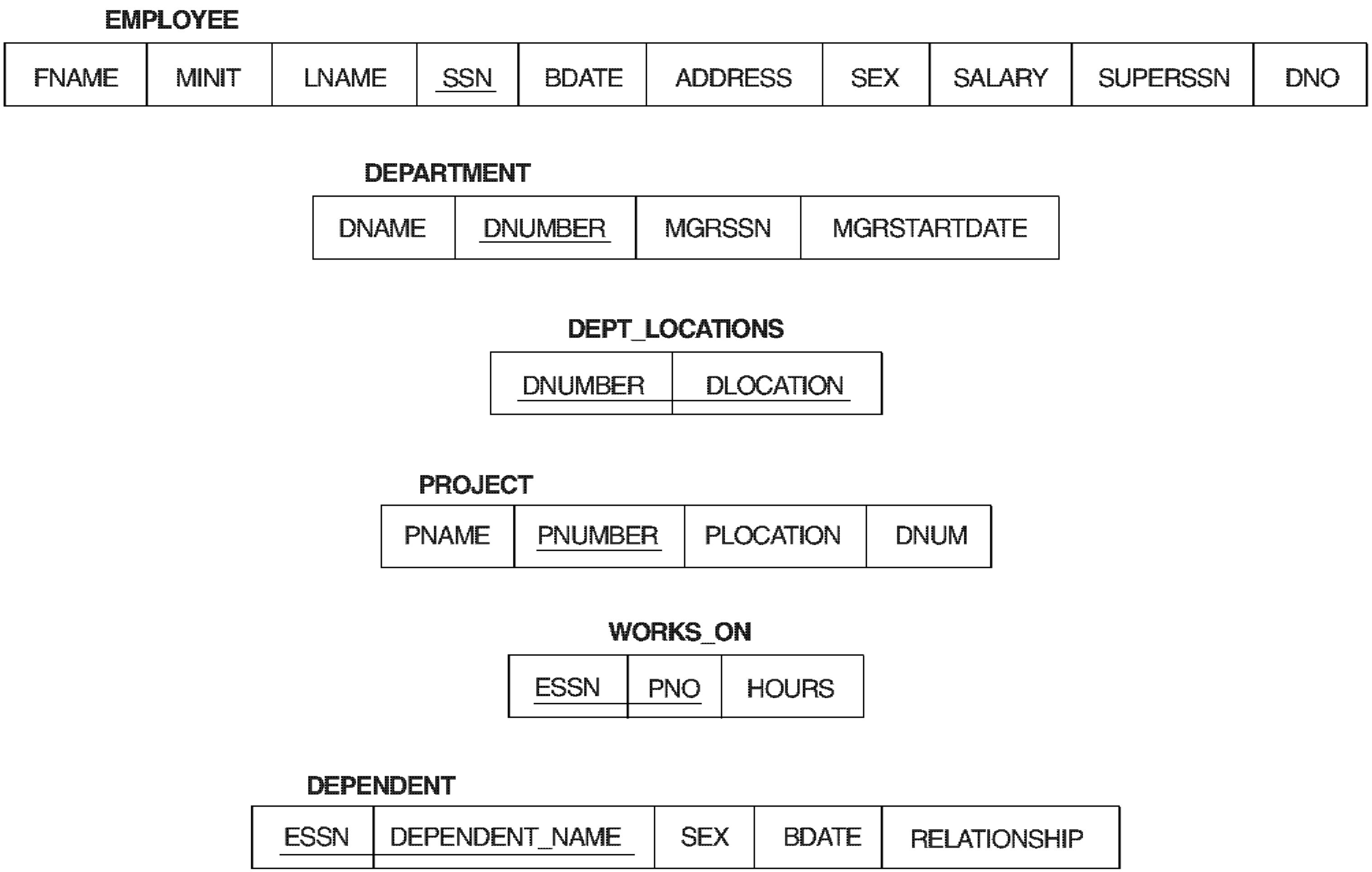
- **Key**
  - Key1 = {LicenseNo}
  - Key2 = {Engine_SerialNo},
  - These are also superkeys.
- **Superkey:** {Engine_SerialNo, Make}
- **Candidate keys:** If a relation has more then one key, we call it **candidate keys**

Figure 7.4 The CAR relation with two candidate keys: LicenseNumber and EngineSerialNumber.



CAR	<u>LicenseNumber</u>	EngineSerialNumber	Make	Model	Year
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Integrity Constraints

- Integrity constraints are specified on DB schema
- They are expected to hold on every valid database state of the schema.
- Integrity constraints ensure accuracy and consistency of data in a relational database.

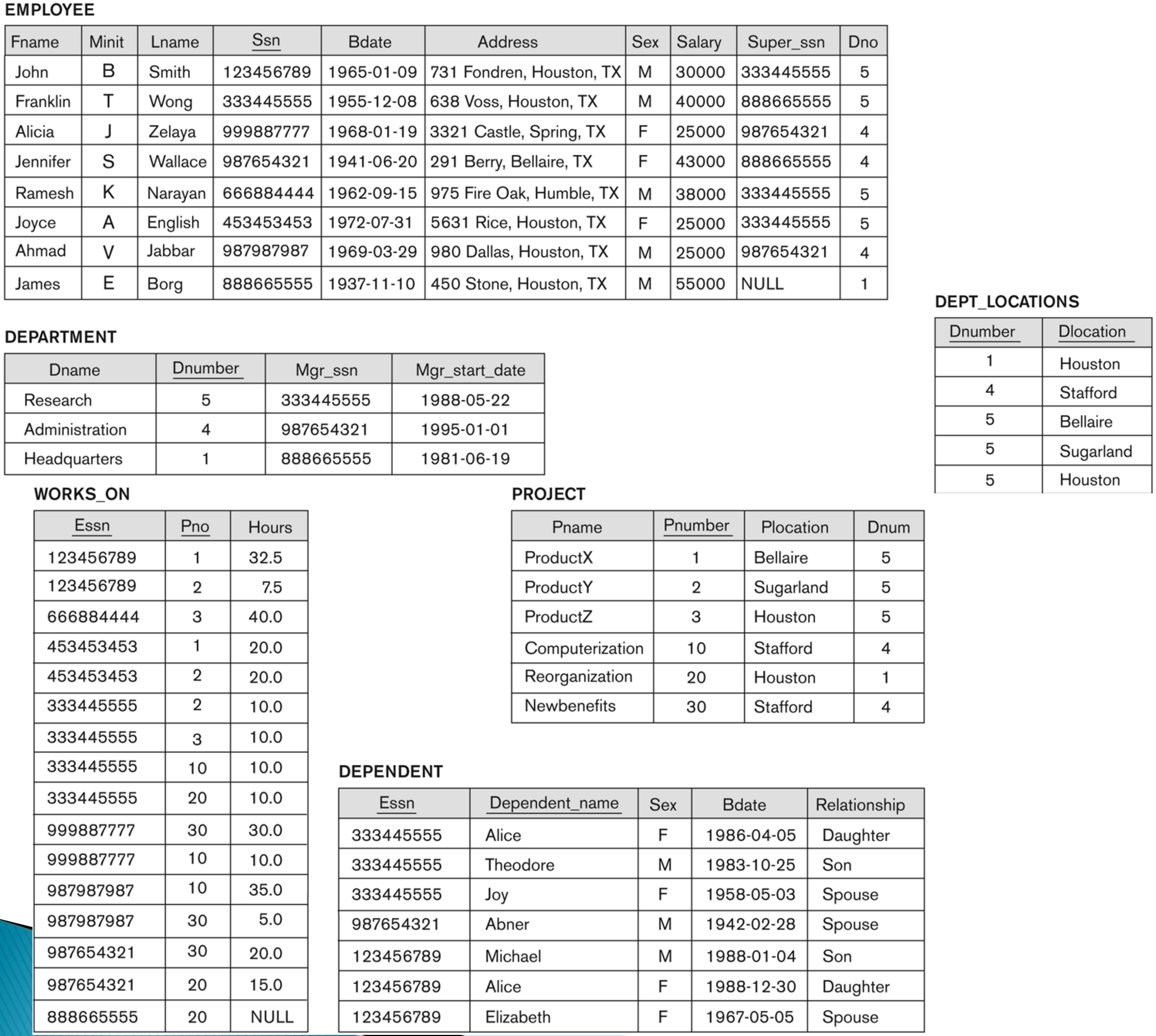
Entity Integrity

- Primary key is used to *identify* the individual tuples so it cannot have null value

Formally,

- Schema S consist of many relations S = {R1, R2, ..., Rn} and State DB{r1,r2,…,rn) where each ri is the state of Ri.
- *Primary key attributes* of each relation schema R in S cannot have null values in any tuple of r(R).

t[PK] ≠ null for any tuple t in r(R)



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

Referential Integrity

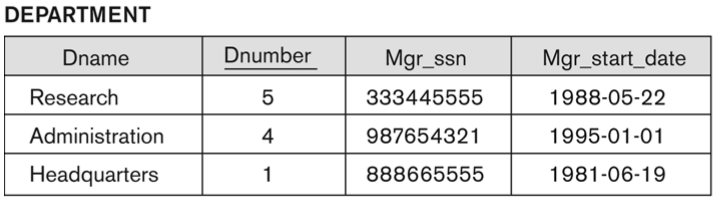
- *A tuple in one relation that refers to another relation must refer to an existing tuple in that relation.*

- This is specified to maintain consistency among tuples in the two relations.

Foreign Key (FK)

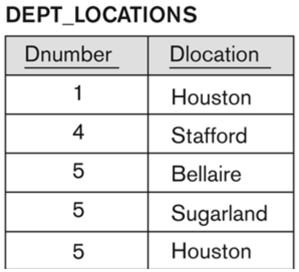
- **A foreign key is a field in a table that matches the primary key column of another table.**

- Set of attributes **FK** in relation R1 is called foreign key of R1 that reference R2 if
  - Attributes in FK in R1 has **same domain** as the attributes in PK of R2
  - **Value of FK** must be an existing PK value in R2 or 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



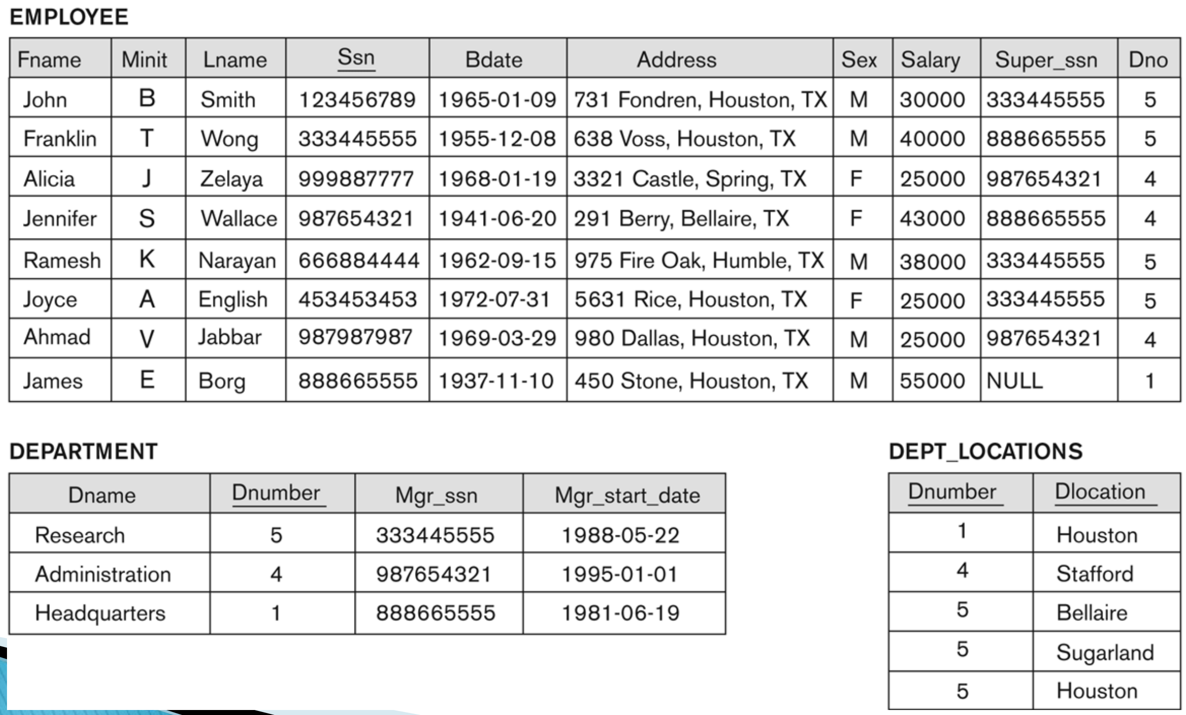
DEPT_LOCATIONS

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

Foreign Key (FK)

- Tuple t1 in R1 is said to reference a tuple t2 in R2 **if t1[FK] = t2[PK].**

  R1 is referencing relation and R2 is reference relation



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

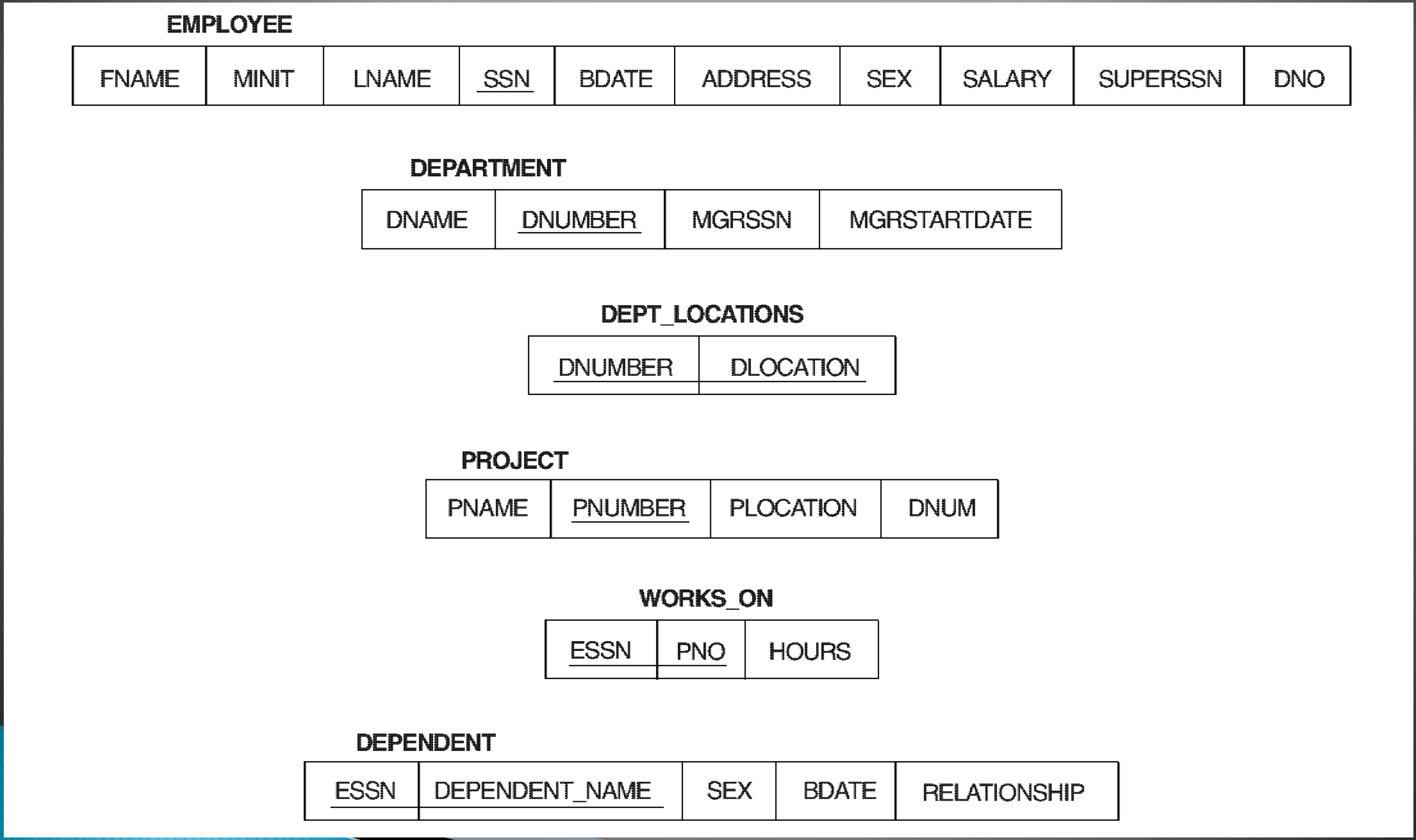
DEPARTMENT

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

Specify Referential Integrity Constraints



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

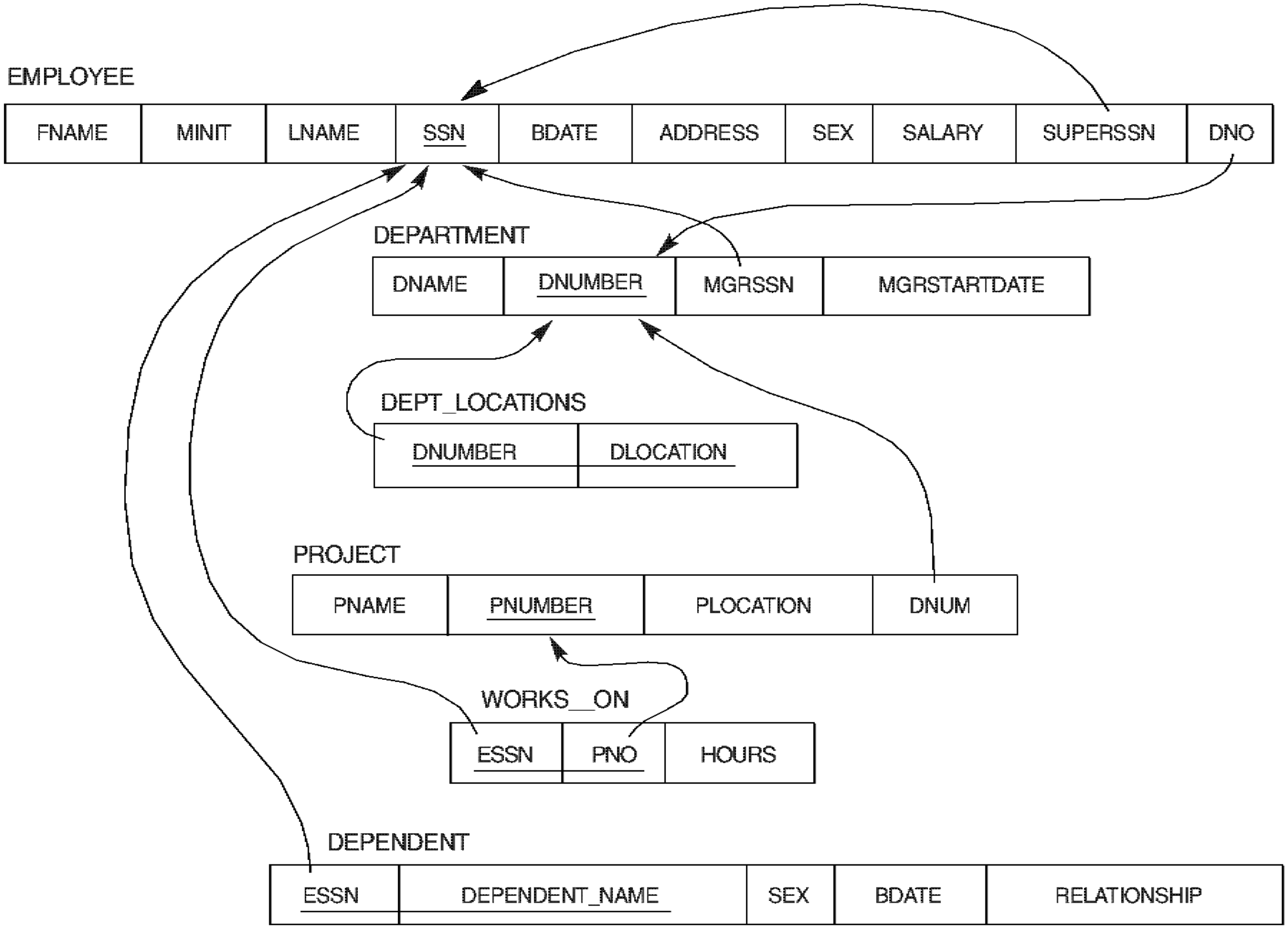
WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

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



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS__ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Foreign Key (FK)

- A foreign key can be a primary key or any of the candidate key in the referenced relation.
- FK requires that uniqueness constraint should hold for the column on which it is defined in referenced relation.
- *Sql server allows specifying FK on a column which has uniqueness constraint set.*

Semantic Integrity Constraints

- Based on application semantics and cannot be expressed by the model schema

- **Example**
  - The max no of hours per employee for all projects he or she works on is 56 hrs per week
  - Employee salary should not exceed supervisor salary

- SQL-99 allows triggers and **ASSERTIONS** to allow for some of these

Operations in RDBMS

Operations in RDBMS are classified as

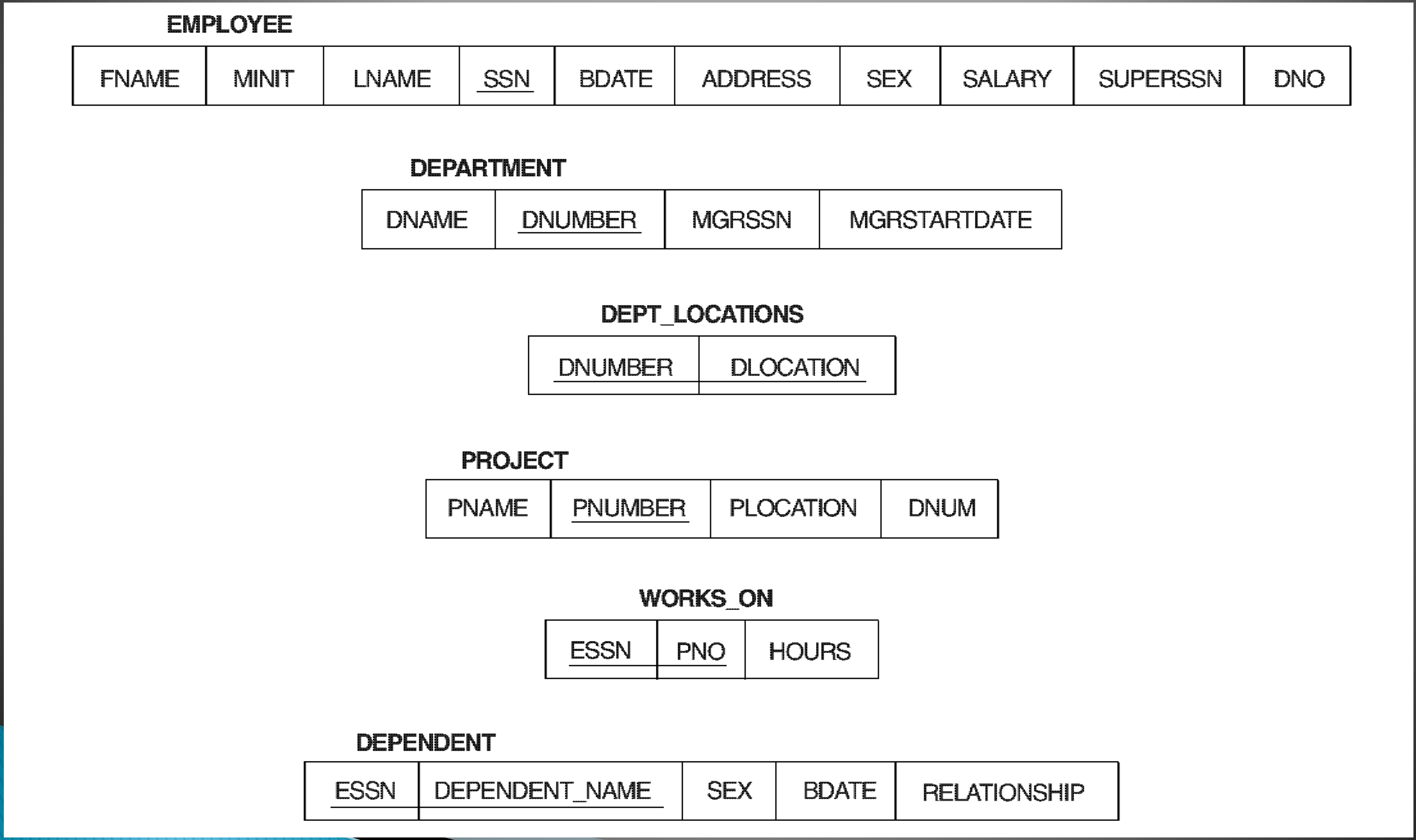
- Retrieval
- Update

- **Update Operations on Relations**
  - INSERT a tuple.
  - DELETE a tuple.
  - MODIFY a tuple.
  - Integrity constraints should not be violated by the update operations.

Operations and Constraints

- **Insert Operation** can violate following constraints
  - Domain constraint
  - Key constraint (if key value already exists)
  - Entity (if PK contains NULL)
  - Referential (FK refers to a tuple that does not exist in referenced relation)
- **Delete operation** can violate
  - Referential Integrity constraint
- **Update operation**
  - that involve no PK or FK cause no problem … only domain constraint are to be check
  - Modifying PK is like deleting a tuple and then inserting another

Company Schema



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

- In case of integrity violation, several actions can be taken:
  - Cancel the operation that causes the violation (REJECT option)
  - Perform the operation but inform the user of the violation
  - Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
  - Execute a user-specified error-correction routine