Relational Data Model

(Part 2)

CSI-406 Database Systems

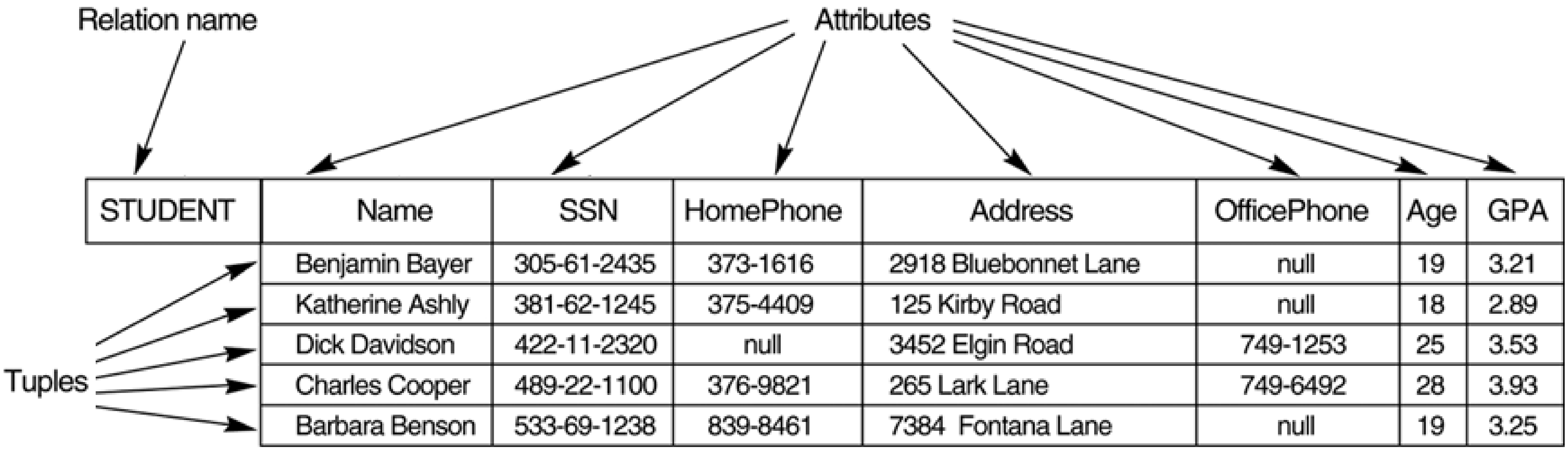
Arfan Shahzad

{ arfanskp@gmail.com }

Relational Data Model
Informal vs. Formal Definitions

Informal Terms	**Formal Terms**
Table	Relation
Column	Attribute/ Domain
Row	Tuple/ Record
Values in a column	(belongs to) Domain
Table Definition	Schema / Intension
Populated Table	State / Extension

Relational Data Model cont…



Relation name

Attributes

Tuples

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Relational Data Model cont…
Characteristics of Relations

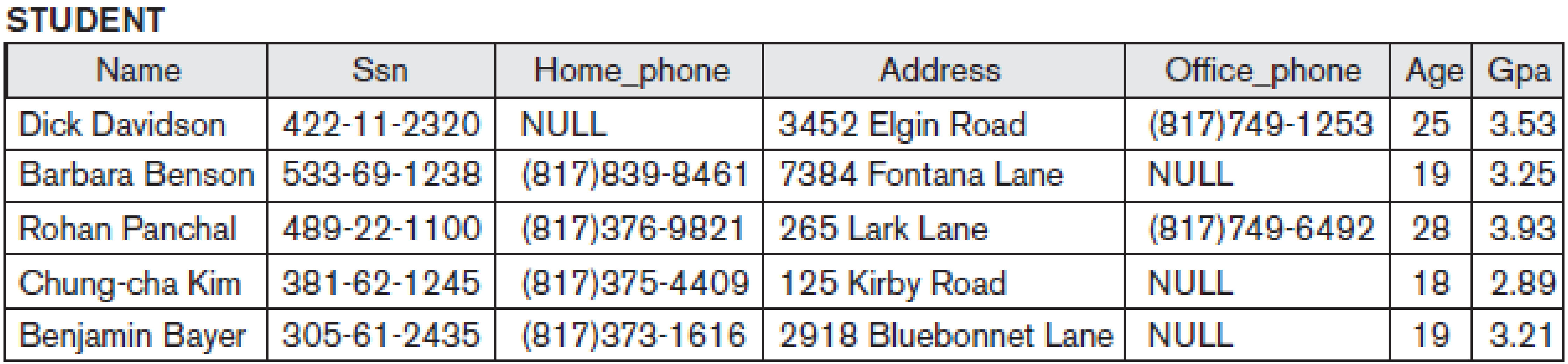
- **Ordering of tuples in a relation r(R):** The tuples are ***not considered to be ordered***, even though they appear to be in the tabular form.

- **Values in a tuple:** All values are considered ***atomic (indivisible)***. A *special null value* is used to represent values that are **unknown** or **inapplicable** to certain tuples.

Relational Data Model cont… Characteristics of Relations

- **Ordering of attributes in a relation schema R (and of values within each tuple):** We will consider the attributes in **$R(A_1, A_2, ..., A_n)$** and the values in **$t=<v_1, v_2, ..., v_n>$** ***to be ordered***. (However, a more general ***alternative definition*** of relation does not require this ordering).
- At a more abstract level, the order of attributes and their values is not important; as long as **the** ***correspondence between attributes and values is maintained***.

Relational Data Model cont… Characteristics of Relations



STUDENT

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

Relational Data Model cont…
Relational Model Notations

- The **degree of a relations** is the ***number of columns*** in a given **relation (table)**.
- A **relation schema**, **R** of **degree n** is denoted by **R (A1, A2, ..., An)**.
- The **uppercase letters Q, R, S** denote **relation schema**.
- The **lowercase letters q, r, s** denote **relation states**.
- The letters **t, u, v** denote **tuples**.

Relational Data Model cont…
Relational Model Notations

- An **attribute A** can be qualified with the **relation name R** to ***which it belongs*** by ***using the dot notation*** **R.A**—for example, **STUDENT.Name** or **STUDENT.Age**.
- This is because ***the same name may be used for two attributes in different relations***.
- However, all attribute names in a particular relation must be distinct.

Relational Data Model cont…
Relational Model Notations

- An **n-tuple t in a relation r(R)** is denoted by **t = <v1, v2, …, vn>**, where **vi** is the value corresponding to attribute **Ai**.
- The following notation refers to component values of tuples:
- Both **t[Ai]** and **t.Ai** (and sometimes **t[i]**) ***refers to the value vi in t for attribute Ai***.

Relational Data Model cont…
Relational Model Notations

- As an example, consider the tuple:
- t = <**‘Barbara Benson’**, **‘533-69-1238’, ‘(817)839-8461’**, **‘7384 Fontana Lane’**, **NULL**, **19**, **3.25**>;
- We have **t[Name] = <‘Barbara Benson’>**, and **t[Ssn, Gpa, Age] = <‘533-69-1238’,3.25, 19>**.

Relational Data Model cont…
Relational Model Constraints

- Constraints are *conditions* that **must hold on *all* valid relation instances**. There are three main types of constraints:

1. **Inherent model-based constraints** or **implicit constraints**.
2. **Schema-based constraints** or **explicit constraints.**
3. **Application-based** or **semantic constraints** or **business rules**

Relational Data Model cont…
Relational Model Constraints

1. Inherent in the data model called inherent model-based constraints or implicit constraints: a relation ***cannot have duplicate tuple*** is inherent constraint.

2. Can be ***directly expressed in schema of the data model***, typically by ***specifying them in the DDL***, called schema-based constraints or explicit constraints **(e.g. Entity integrity** constraints and **Referential integrity** constraints)

3. ***Cannot be directly expressed in the schemas of the data*** **model**, and hence **must be expressed and enforced by the application programs**, called application-based or semantic constraints or business rules.

Relational Data Model cont…
Relational Model Constraints

- Elaborating upon **schema-based constraints**:

1. **Domain Constraints**
2. **Key Constraints**
3. **Entity Integrity Constraint**
4. **Referential Integrity Constraint**

Relational Data Model cont…
Domain Constraints

- Each attribute value must be either **null** (which is really a *non-value*) or ***drawn from the domain of that attribute***.
- Note that some DBMS's allow you to impose the **not null** constraint upon an attribute, which is to say that **no tuple (in the relevant relation) is allowed to have the (non-)value null in that attribute**.

Relational Data Model cont…
Key Constraints

- **Super key** of **R:** A set of attributes SK of R such that **two tuples** ***in any valid relation instance r(R)*** **will never have the same value for SK**. That is, for any distinct tuples t1 and t2 in r(R), **t1[SK] ≠ t2[SK]**.
- **Key** of **R**: **A "minimal" superkey;** that is, a superkey from which ***we cannot remove any attributes and still have the uniqueness constraint***.
- **Candidate Key:** A relation may have two or more **super keys**, ***all these super keys will called as candidate keys (to become primary key)***.

Relational Data Model cont…
Key Constraints

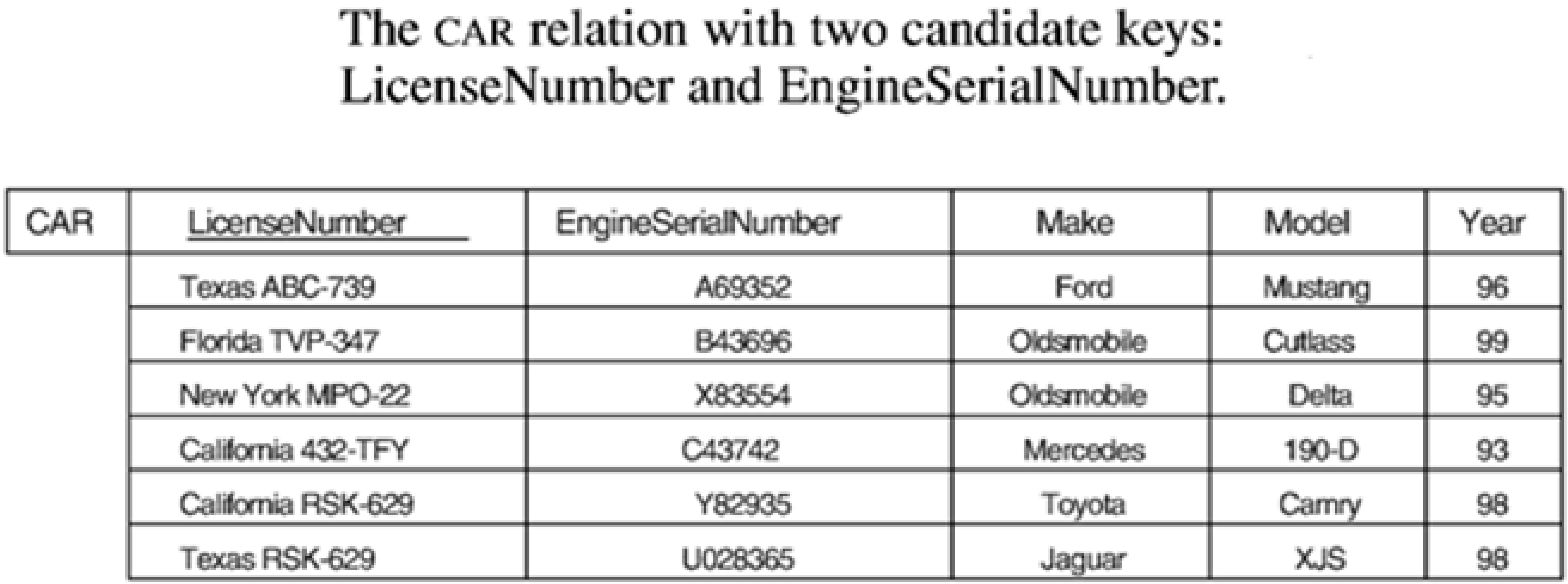
- **Primary key:** **Successful candidate key** will be called **primary key**. In schema, the **primary key** attributes are **underlined**.

- **Example**: The CAR relation schema:

- **CAR** (State, Reg#, SerialNo, Make, Model, Year):

- Has two keys Key1 = {State, Reg#}, Key2 = {SerialNo, Make}, which are also superkeys. {SerialNo, Make} is a superkey but *not* a key.

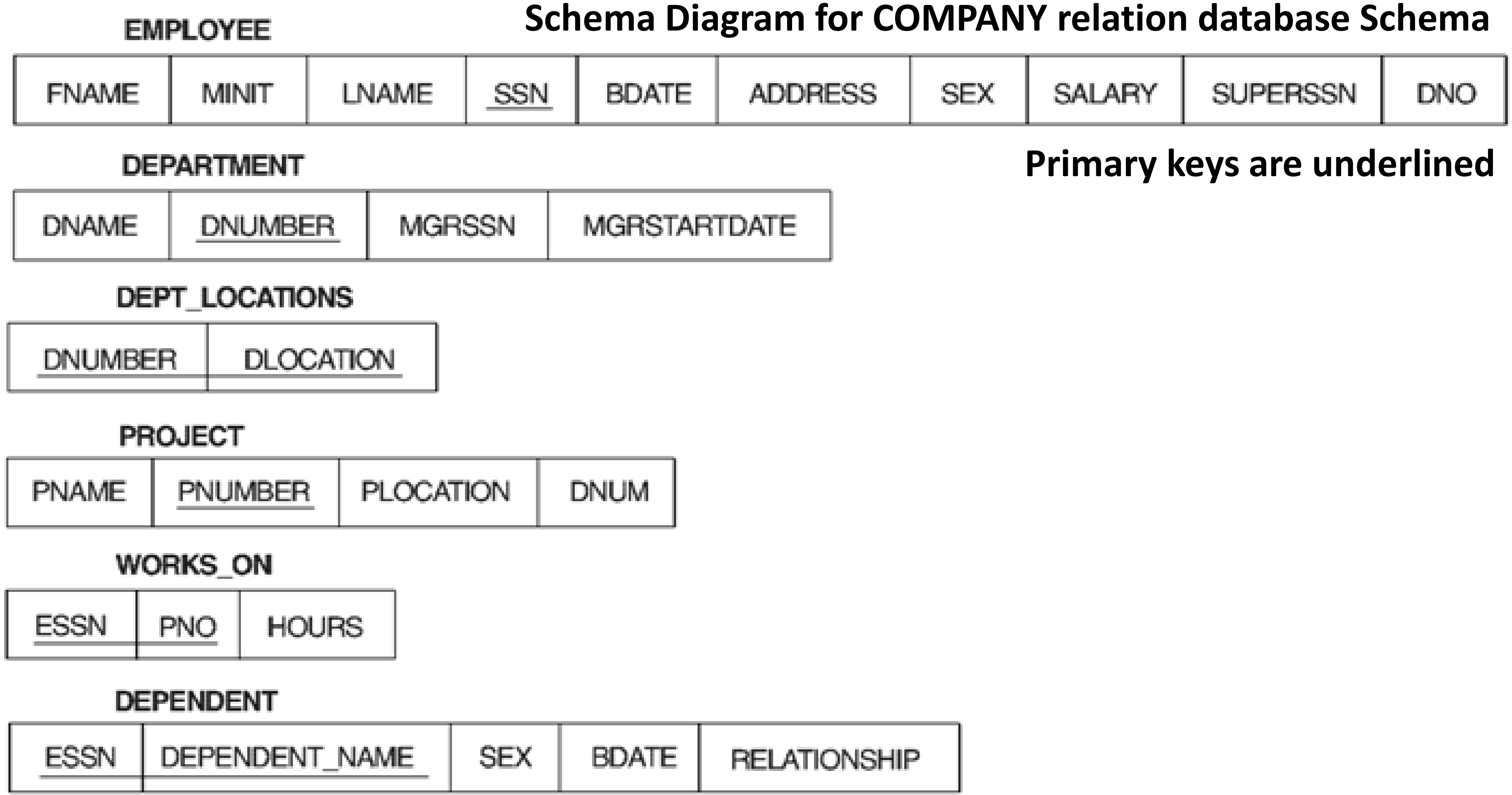
Relational Data Model cont…
Key Constraints



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

CAR	LicenseNumber	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

Relational Data Model cont…
Key Constraints



Schema Diagram for COMPANY relation database Schema

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

Relational Data Model cont…
Entity Integrity Constraint

- **Relational Database Schema**: A set S of relation schemas that belong to the same database. S is the *name* of the **database**. S = {R_1, R_2, ..., R_n}
- **Entity Integrity**: The ***primary key attributes*** PK of each relation schema **R** in **S** ***cannot have null values in any tuple of r(R)***. This is because primary key values are used to *identify* the individual tuples.
- **t[PK] ≠ null for any tuple t in r(R)**

Relational Data Model cont…
Entity Integrity Constraint

- **Note:** Other attributes of R may be similarly constrained to disallow null values, even though they are not members of the primary key (until, unless specified).

Relational Data Model cont…
Referential Integrity Constraint

- A special constraint involving *two* relations, used to specify a ***relationship*** **among tuples in two relations**: the **referencing relation** and the **referenced relation**.
- The previous constraint (i.e. Entity Integrity) involve a *single* relation.

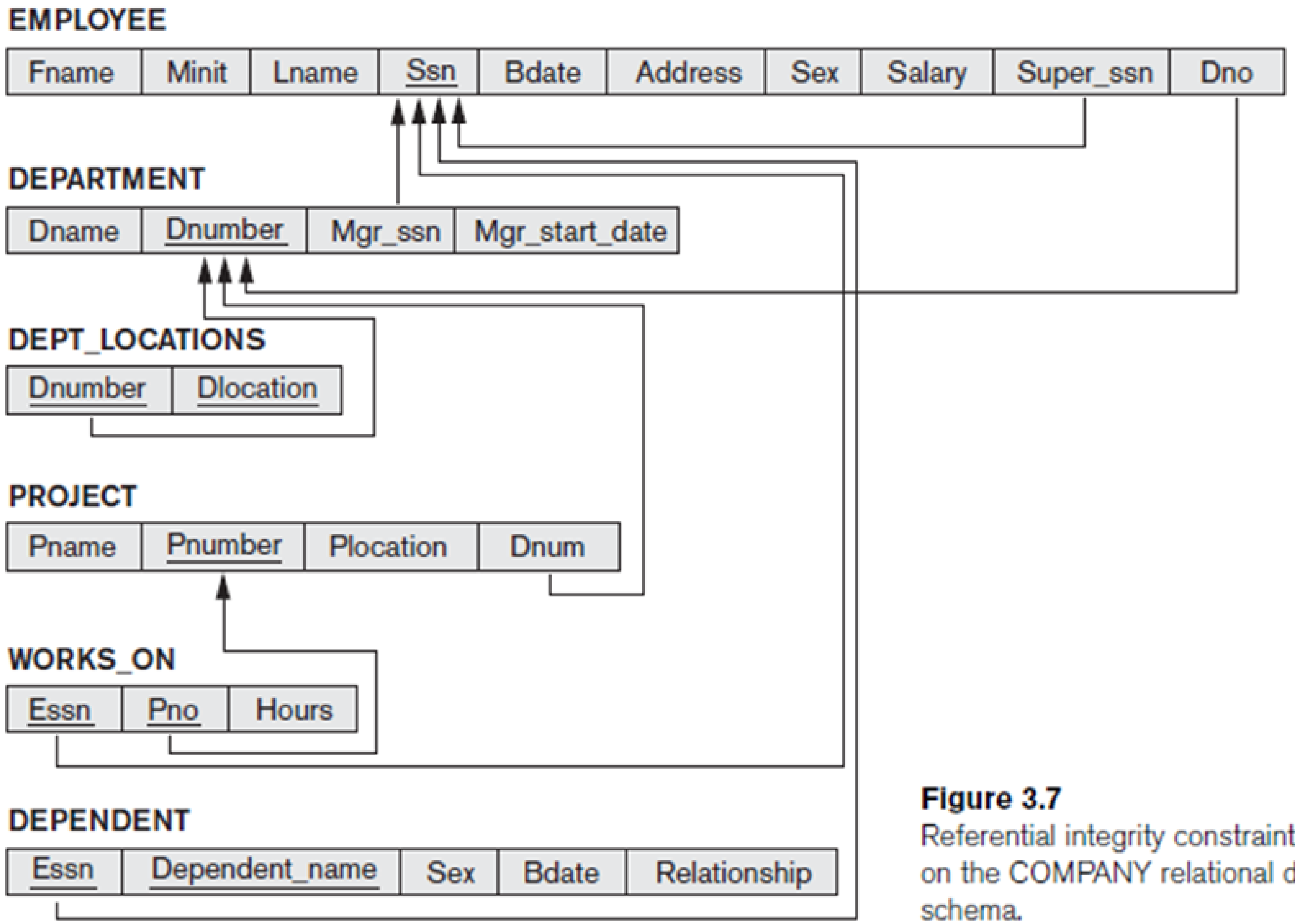
Relational Data Model cont…
Referential Integrity Constraint

- Tuples in the ***referencing relation*** R_1 have attributes **FK** (called **foreign key** attributes) that **reference** the **primary key attributes PK** of the ***referenced relation*** R_2.
- A tuple **t_1 in R_1** is said to **reference** a tuple **t_2 in R_2** if **t_1[FK] = t_2[PK]**.
- A referential integrity constraint can be displayed in a relational database schema **as a directed arc from R_1.FK to R_2**.

Relational Data Model cont… Referential Integrity Constraint

- **Statement of the constraint:** The value in the foreign key column (or columns) FK of the **referencing relation** R_1 can be either:
- (1) Same as a value of an **existing primary key value** of the corresponding primary key PK in the **referenced relation** $R_{2,}$, or..
- (2) a null.
- In case (2), the FK in R_1 should not be a part of its own primary key.

Relational Data Model cont…
Referential Integrity Constraint



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Relational Data Model cont…
Other Types of Constraints

- **Semantic Integrity Constraints:** based on **application semantics** and cannot be expressed by the data model (or DDL). E.g.,
  - The max. no. of hours per employee for all projects (he or she works on) is 56 hrs per week
  - Salary of a supervisee cannot be greater than that of her/his supervisor
  - Salary of an employee cannot be lowered
  - A department manager cannot be less than 35 years old

Class Assignment

- Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course:
- STUDENT(SSN, Name, Major, Bdate)
- COURSE(Course#, Cname, Dept)
- ENROLL(SSN, Course#, Quarter, Grade)
- BOOK_ADOPTION(Course#, Quarter, Book_ISBN)
- TEXT(Book_ISBN, Book_Title, Publisher, Author)
- **Draw a relational schema diagram specifying the foreign keys for this schema.**