

Fundamentals of Database Systems

Chapter 4:

The Relational Data Model and Relational Database Constraints





Overview

- ◎ The **relational data model** is based on having a set of relations with relationships between them.
- ◎ A relation looks like a **table** of values.
- ◎ A relation has columns (formally called attributes).
- ◎ A relation typically also contains a **set of rows (also called tuples)**.
- ◎ The data elements in each **row** represent certain facts that correspond to a real-world **entity** or **relationship**



Example #1 of a Relation

Citizens

Name	National_ID	Date_of_Birth
Sami Fahad	1078114611	01/04/1990
Ahmad Saad	1051111654	05/04/1991
Manal Khalid	105551231	09/11/1989



Example #1 of a Relation

Relation name
→ **Citizens**

Attributes

Name	National_ID	Date_of_Birth
Sami Fahad	1078114611	01/04/1990
Ahmad Saad	1051111654	05/04/1991
Manal Khalid	105551231	09/11/1989

tuples



Example #2 of a Relation

Diagram illustrating the structure of a relation (table) named **STUDENT**.

The relation is defined by its **Attributes** (columns) and **Tuples** (rows).

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	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
Rohan Panchal	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

Figure 5.1

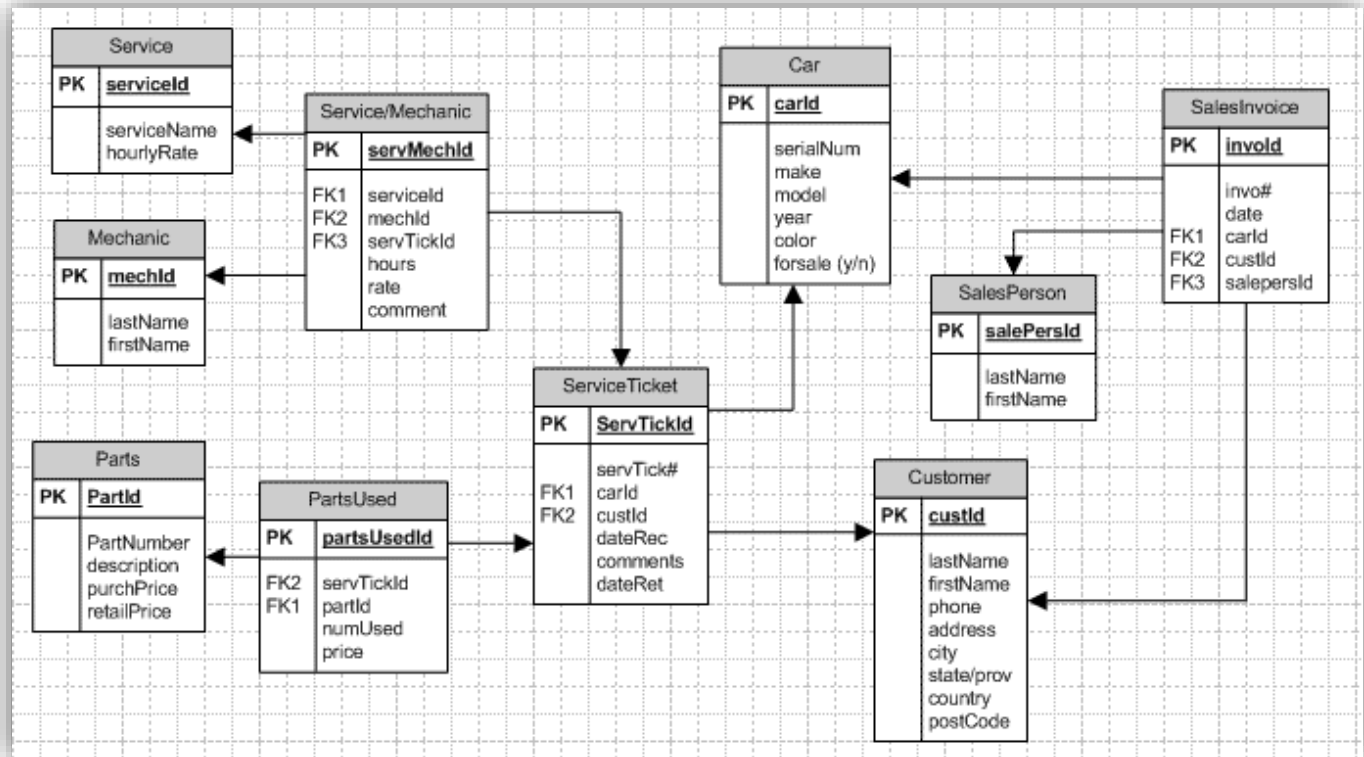
The attributes and tuples of a relation STUDENT.



Key of a relation

- Each relation must have a key
- **Key:** attribute which value in each row must be unique
- In the **CITIZENS** table, **National_ID** is the key
- In the **STUDETN** table, **SSN** is the key
- Key for table with *cars dealership* information?
- Key for table with *flights* information?

Key for tables with cars dealership information?





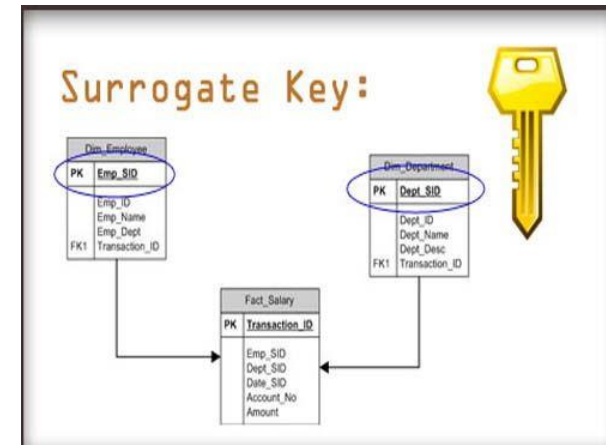
Key of a relation

- **Primary key (PK):** An attribute that identify all table records
- A primary key **must** only contain **unique values**
- It **cannot** have **null values**.
- If relation has more than one key, you can pick one as PK
- A PK can be two attributes combined (student id and section number)
- **Surrogate key:** Sometimes row-ids or *sequential numbers* are assigned as keys to identify the rows in a table

unique value for each row of data

Can not contain NULL values

should not be altered frequently





Schema

- The **Schema** (or description) of a Relation:
- Denoted by $R (A_1, A_2, \dots A_n)$
- R is the **name** of the relation
- The **attributes** of the relation are $A_1, A_2, \dots A_n$

- **Example:** CUSTOMER (Cust_id, Cust_name, Address, Phone#)
- CUSTOMER is the relation name
- Defined over the four attributes: Cust_id, Cust_name, Address, Phone#

- Each attribute has a domain or a set of valid values.
- For example, the domain of *Cust_id* is 6 digit numbers.



Schema (example)

```
create table emp(  
  empno      number(4,0),  
  ename      varchar2(10),  
  job        varchar2(9),  
  mgr        number(4,0),  
  hiredate   date,  
  sal        number(7,2),  
  comm       number(7,2),  
  deptno     number(2,0),  
  constraint pk_emp primary key (empno),  
  constraint fk_deptno foreign key (deptno) references dept (deptno)  
)
```

Create the EMP table which has a foreign key reference to the DEPT table. The foreign key will require that the DEPTNO in the EMP table exist in the DEPTNO column in the DEPT table.



Data Types (example)

Data types are used to represent the nature of the data that can be stored in the database table.

Data types mainly classified into three categories for every database.

- String Data types
- Numeric Data types
- Date and time Data types

	Column Name	Data Type	Allow Nulls
🔑	PostID	int	<input type="checkbox"/>
	StatusID	int	<input type="checkbox"/>
	OwnerID	int	<input type="checkbox"/>
	PostCategoryID	int	<input type="checkbox"/>
	ApprovalUserID	int	<input checked="" type="checkbox"/>
	VisitedCounter	int	<input type="checkbox"/>
	Title	nvarchar(100)	<input type="checkbox"/>
	ContentText	nvarchar(MAX)	<input type="checkbox"/>
	CreatedDate	datetime2(7)	<input type="checkbox"/>
	ModifiedDate	datetime2(7)	<input checked="" type="checkbox"/>
	ExpireDate	datetime2(7)	<input type="checkbox"/>
	ApprovalDate	datetime2(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>



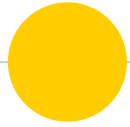
Domain

- A **domain** defines a structure or format that all values of an attribute must follow.
- **For example**, all phone numbers in Tunisia have to be 8 digits long and the first digit has to be zero.
- Domain also has a data-type
- **For example**, students' names must be a string
- A special **null** value is used to represent values that are unknown or not available or inapplicable.



Definition Summary

Informal Terms	Formal Terms
Table	Relation
Column	Attribute
All possible Column Values	Domain
Row	Tuple
Table Definition	Schema of a Relation
Populate Table	State of the Relation



Part 2: Constraints



Constraints

- Constraints determine which values are allowed and which are not in the database.
- Constraints are **conditions** that must hold on **all** valid relation states.
- **Three main types of relational integrity constraints:**
 1. Key constraints
 2. Entity integrity constraints
 3. Referential integrity constraints



Key Constraints

- **Candidate key:** A super key with no repeated attribute is called candidate key.
- Relational schema may have more than one key.
- **Candidate key:** an attribute that satisfies the \mathcal{PK} conditions.

CAR

<u>License_number</u>	Engine_serial_number	Make	Model
Texas ABC-739	A69352	Ford	Mustang
Florida TVP-347	B43696	Oldsmobile	Cutlass
New York MPO-22	X83554	Oldsmobile	Delta
California 432-TFY	C43742	Mercedes	190-D
California RSK-629	Y82935	Toyota	Camry
Texas RSK-629	U028365	Jaguar	XJS

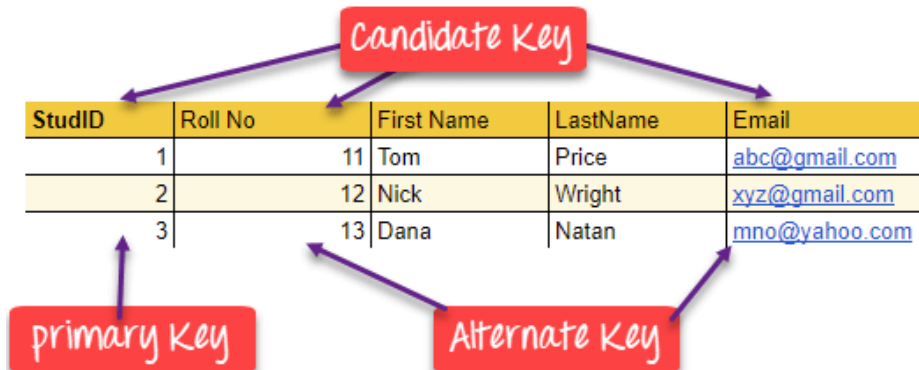


Key Constraints

- **Candidate key:** You can select either *license number* or the *engine serial number* as the primary key.
- **Key constraint:** Ensuring that all rows in a relation satisfy the rules of the PK.

● **Properties of Candidate key:**

- It must contain unique values
- Candidate key may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table





Displaying PKs in a database schema

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Figure 5.5
Schema diagram for
the COMPANY
relational database
schema.



Referential Integrity Constraint

- A constraint involving two relations (two tables).
- **Foreign key:** An attribute in R_1 where any value of the attribute must reference a primary key in R_2 .



Referential Integrity Constraint

Player

<u>Online_ID</u>	<u>Date_of_Birth</u>	Country
Maha1	03/12/1991	Saudi Arabia
Hassan99	09/01/1992	Egypt
Nassir 3	01/07/1995	Saudi Arabia

Game

<u>Game_ID</u>	Player_1	Player_2	Winner
1	Maha1	Hassan99	Hassan99
9	Hassan99	Nassir 3	Nassir 3
12	Nassir 3	Maha1	Maha1
43	Maha1	Nassir 3	Maha1
5	Hassan99	Nassir 3	Hassan99
10	Nassir 3	Maha1	Maha1

League

<u>League_ID</u>	<u>Online_ID</u>	Status
0001	Maha1	Active
0002	Hassan99	Active
0003	Maha1	Inactive



Referential Integrity Constraint

PKs:

1. Online_ID (Player)
2. Game_ID (Game)
3. League_ID (League)

FKs:

1. Player_1 and Player_2 are referencing Online_ID in the relation Player
2. Online_ID in League is referencing Online_ID in Player

Player

<u>Online_ID</u>	<u>Date_of_Birth</u>	<u>Country</u>
Mahal	03/12/1991	Saudi Arabia
Hassan99	09/01/1992	Egypt
Nassir_3	01/07/1995	Saudi Arabia

Game

<u>Game_ID</u>	<u>Player_1</u>	<u>Player_2</u>	<u>Winner</u>
1	Mahal	Hassan99	Hassan99
9	Hassan99	Nassir_3	Nassir_3
12	Nassir_3	Mahal	Mahal
43	Mahal	Nassir_3	Mahal
5	Hassan99	Nassir_3	Hassan99
10	Nassir_3	Mahal	Mahal

League

<u>League_ID</u>	<u>Online_ID</u>	<u>Status</u>
0001	Mahal	Active
0002	Hassan99	Active
0003	Mahal	Inactive



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) a value of an existing primary key value of a corresponding primary key PK in the referenced relation R_2 , or
 - (2) a null
- An FK can also be a PK
- An FK cannot be null if it's also a PK



Displaying a relational **database schema and its constraints**

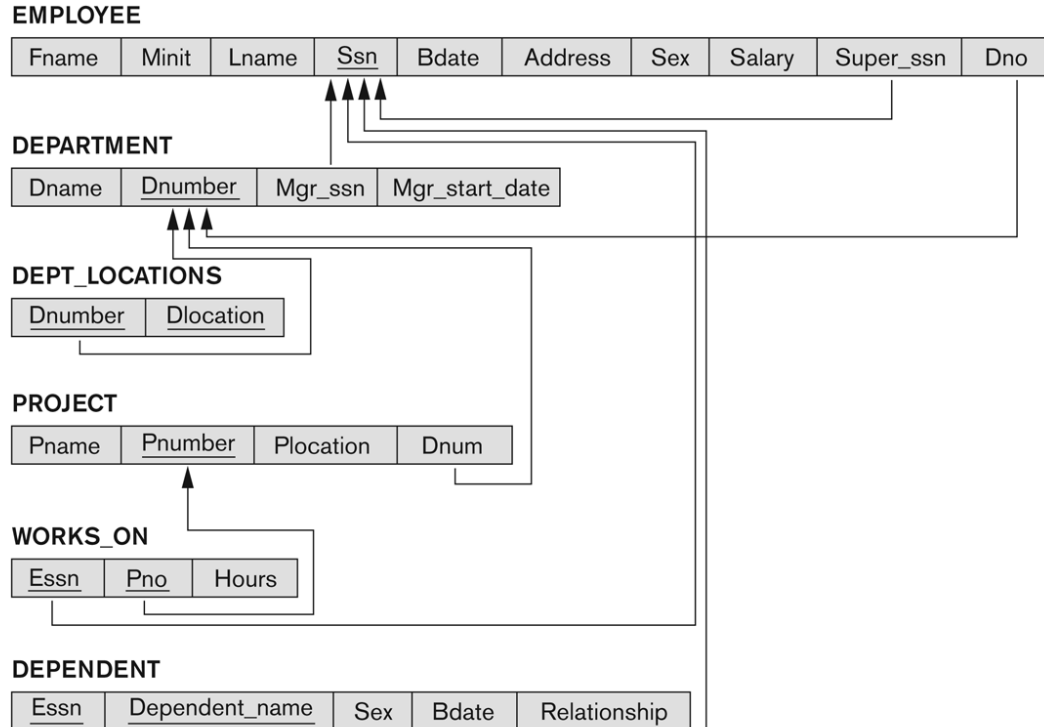
- Each **relation schema** can be displayed as a row of attribute names
- The **name** of the relation is written above the attribute names
- The **primary key** attribute (or attributes) will be **underlined**
- A **foreign key** (referential integrity) constraints is displayed as a directed arc (arrow) from the foreign key attributes to the referenced table



Referential Integrity Constraints Example

Figure 5.7

Referential integrity constraints displayed on the COMPANY relational database schema.





Entity Integrity

- ⦿ The primary key attributes PK of each relation schema R cannot have null values in any tuple of $r(R)$.
- ⦿ This is because primary key values are used to *identify* the individual tuples.
- ⦿ If PK has several attributes, null is not allowed in any of these attributes