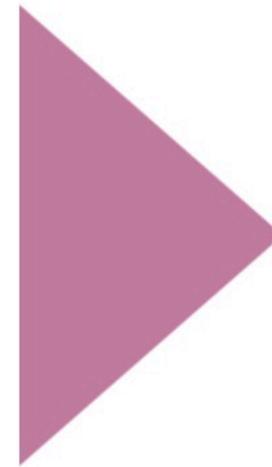


Relational DB & SQL

Session 1





Fundamentals of RDBMS

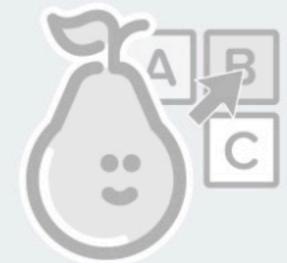


I've completed the pre-class?



Students choose an option

Pear Deck Interactive Slide
Do not remove this bar

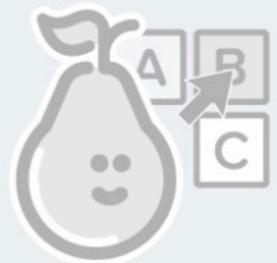


No Multiple Choice Response
You didn't answer this question

I've installed and started SQL Server Management Studio.

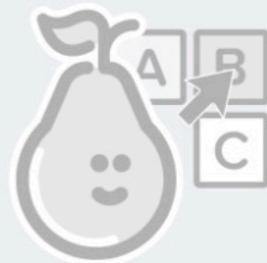


Pear Deck Interactive Slide
Do not remove this bar



No Multiple Choice Response
You didn't answer this question

I've created SampleRetail database and executed a query.



No Multiple Choice Response
You didn't answer this question



Students choose an option

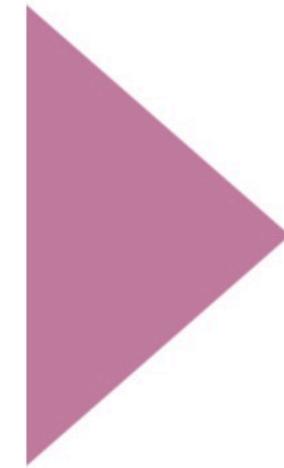
REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar



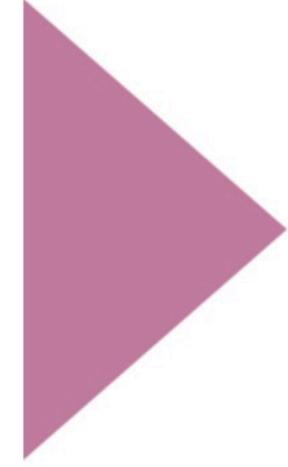
Table of Contents

- ▶ Relational Database Concepts
- ▶ Integrity Rules
- ▶ Key Constraints
- ▶ Normalization



Relational Database Concepts





- ★ Data
- ★ Table
- ★ Relations
- ★ Relationships
- ★ Domain
- ★ Column
- ★ Row
- ★ Normalization
- ★ ERD
- ★ Integrity Rules
- ★ Constraints
- ★ Data Types



► Data (Structured)

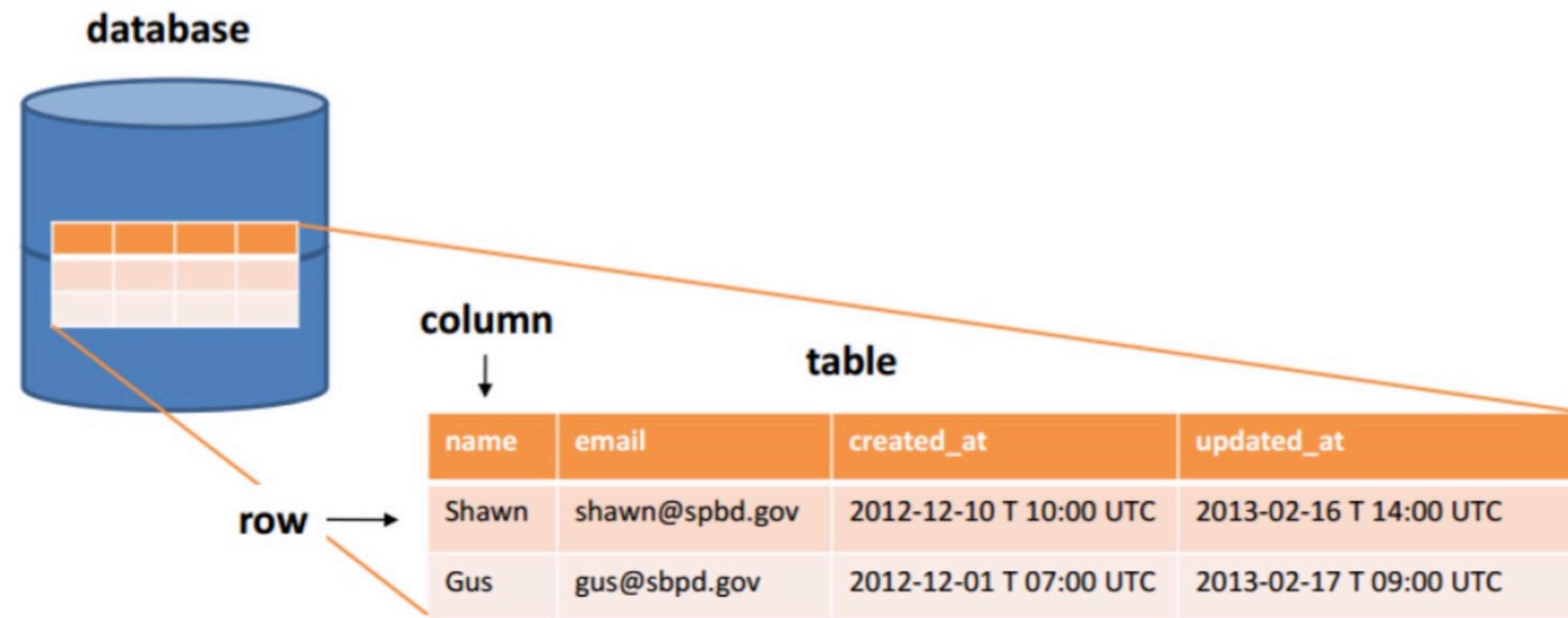


| <u>Std ID</u> | First Name | Last Name | Email |
|---------------|------------|-----------|--|
| 1109 | John | Price | jp@gmail.com |
| 1401 | Bran | Angel | bw@gmail.com |
| 1756 | Andrew | Natan | cn@gmail.com |
| 1945 | Tom | Black | tb@gmail.com |
| 1987 | Nick | Norris | nn@gmail.com |

► Database Properties



A database has the following properties:



► What is Metadata?

A set of data that describes and gives information about data.



► Data vs Metadata



| <u>Std ID</u> | First Name | Last Name | Email |
|---------------|------------|-----------|--|
| 1109 | John | Price | jp@gmail.com |
| 1401 | Bran | Angel | bw@gmail.com |
| 1756 | Andrew | Natan | cn@gmail.com |
| 1945 | Tom | Black | tb@gmail.com |
| 1987 | Nick | Norris | nn@gmail.com |

► Domain



A domain is the original sets of atomic values used to data modeling.

For example:

- The domain of Marital Status has a set of possibilities: Married, Single, Divorced.
- The domain of First Name is the set of character strings that represents names of people.

► Domain Example



| country_id | region_id | name | area | national_day | code2 | code3 |
|------------|-----------|-------------|------------|--------------|-------|-------|
| 1 | 1 | Aruba | 193.00 | NULL | AW | ABW |
| 2 | 2 | Afghanistan | 652090.00 | 1999-08-19 | AF | AFG |
| 3 | 3 | Angola | 1246700.00 | 1975-11-11 | AO | AGO |
| 4 | 1 | Anguilla | 96.00 | 1967-05-30 | AI | AIA |
| 32 | 1 | Barbados | 430.00 | 1966-11-30 | BB | BRB |
| 34 | 2 | Bhutan | 47000.00 | NULL | BT | BTN |



list of
characters list of
numeric values



► Column



The principal storage units are called columns.

| Column1 | Column2 | Column3 | Column4 | Column5 | Column6 | Column7 |
|------------|-----------|-------------|------------|--------------|---------|---------|
| country_id | region_id | name | area | national_day | code2 | code3 |
| 1 | 1 | Aruba | 193.00 | NULL | AW | ABW |
| 2 | 2 | Afghanistan | 652090.00 | 1999-08-19 | AF | AFG |
| 3 | 3 | Angola | 1246700.00 | 1975-11-11 | AO | AGO |
| 4 | 1 | Anguilla | 96.00 | 1967-05-30 | AI | AIA |
| 32 | 1 | Barbados | 430.00 | 1966-11-30 | BB | BRB |
| 34 | 2 | Bhutan | 47000.00 | NULL | BT | BTN |

► Record



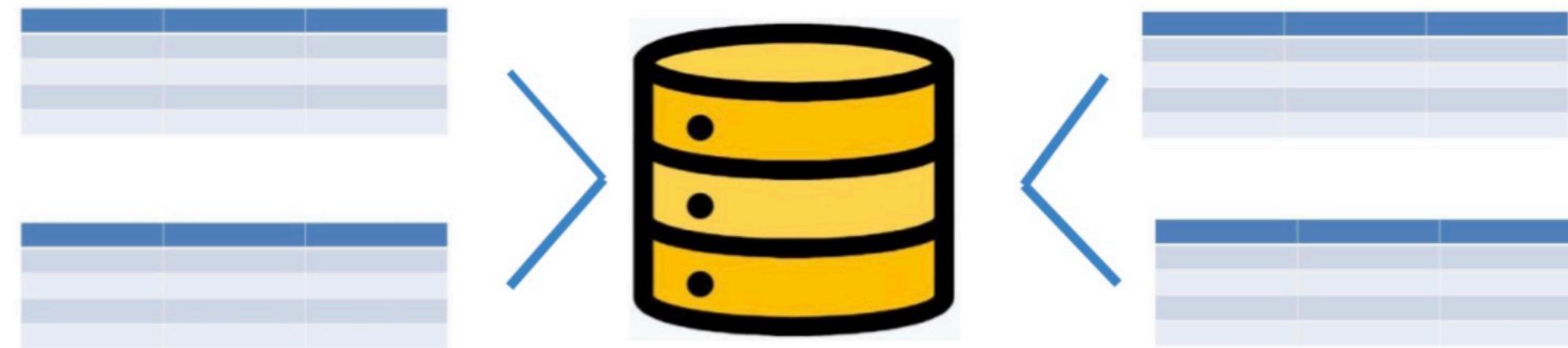
Records contain fields that are related, such as a customer or an employee. A tuple is another term used for record.

| | country_id | region_id | name | area | national_day | code2 | code3 |
|---------|------------|-----------|-------------|------------|--------------|-------|-------|
| Record1 | 1 | 1 | Aruba | 193.00 | NULL | AW | ABW |
| Record2 | 2 | 2 | Afghanistan | 652090.00 | 1999-08-19 | AF | AFG |
| Record3 | 3 | 3 | Angola | 1246700.00 | 1975-11-11 | AO | AGO |
| Record4 | 4 | 1 | Anguilla | 96.00 | 1967-05-30 | AI | AIA |
| Record5 | 32 | 1 | Barbados | 430.00 | 1966-11-30 | BB | BRB |
| Record6 | 34 | 2 | Bhutan | 47000.00 | NULL | BT | BTN |

► Table



A database is composed of multiple tables and each table keep the data.



► Table Properties - 1

- A table has **a distinct name** from all other tables in the database.
- There are **no duplicate rows**; each row is distinct.
- Entries in **columns are atomic**. The table does not contain repeating groups or multivalued attributes.
- Entries are from **the same domain** based on their **data type** including:
 - number (numeric, integer, float, smallint,...)
 - character (string)
 - date
 - logical (true or false)

► Table Properties - 2



- Operations combining different data types are disallowed.
- Each attribute has a **distinct name**.
- The sequence of columns is insignificant.
- The sequence of rows is insignificant.

► Relation



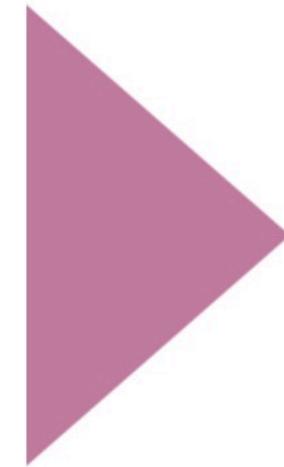
A relation, also known as a table or file, is **a subset of the Cartesian product** of a list of domains characterized by a name.

| countries | | | | | | |
|------------|-----------|-------------|------------|--------------|-------|-------|
| country_id | region_id | name | area | national_day | code2 | code3 |
| 1 | 1 | Aruba | 193.00 | NULL | AW | ABW |
| 2 | 2 | Afghanistan | 652090.00 | 1999-08-19 | AF | AFG |
| 3 | 3 | Angola | 1246700.00 | 1975-11-11 | AO | AGO |
| 4 | 1 | Anguilla | 96.00 | 1967-05-30 | AI | AIA |
| 32 | 1 | Barbados | 430.00 | 1966-11-30 | BB | BRB |
| 34 | 2 | Bhutan | 47000.00 | NULL | BT | BTN |

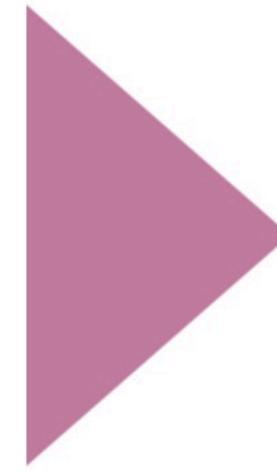
► What is Relational Data Model?



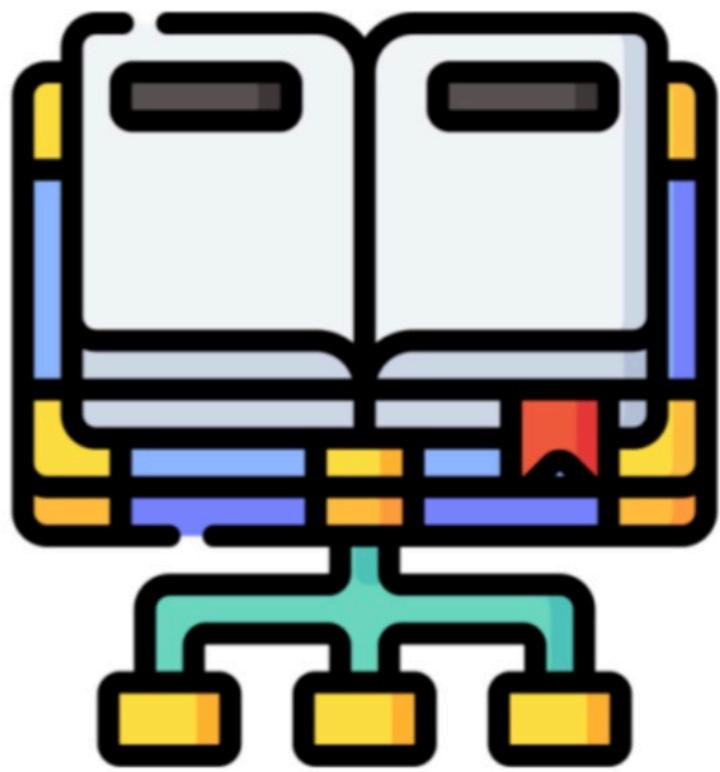
The relational data model describes the world as “a collection of **interrelated relations** (or tables).”



Let's look up Sample Retail DB



Integrity Rules



►Domain Integrity



Domain restricts the values of attributes in the relation and is a constraint of the relational model.

We need more specific ways to state what data values are or are not allowed and which format is suitable for an attribute.

| Std_ID | First Name | Last Name | Email |
|---------|------------|-----------|--|
| 1109 | John | Price | jp@gmail.com |
| 1401 | Bran | Angel | bw@gmail.com |
| 1756 | Andrew | Natan | cn@gmail.com |
| 1945 | Tom | Black | tb@gmail.com |
| 1987abc | Nick | Norris | nn@gmail.com |

This value is out of domain (not INTEGER) so it is not acceptable.



► Entity integrity



To ensure entity integrity, it is required that every table have a primary key. Neither the PK nor any part of it can contain null values.

| <u>Std_ID</u> | First Name | Last Name | Email |
|---------------|------------|-----------|--|
| 1109 | John | Price | jp@gmail.com |
| 1401 | Bran | Angel | bw@gmail.com |
| 1756 | Andrew | Natan | cn@gmail.com |
| 1945 | Tom | Black | tb@gmail.com |
| | Nick | Norris | nn@gmail.com |



This value cannot be NULL as we will
not be able to identify students
uniquely



► Referential integrity

Referential integrity requires that a **foreign key** must have a matching primary key or it must be null.

This constraint is specified between two tables (parent and child); it maintains the **correspondence** between rows in these tables.

It means the **reference** from a row in one table to another table must be valid.

| Crs_ID | CourseNo | Name | Hours |
|--------|----------|-----------|-------|
| 2034 | CS305 | Database | 4 |
| 1263 | CS267 | Comp Arch | 4 |

| Std_ID | Crs_ID | Semester | Grade |
|--------|--------|----------|-------|
| 1401 | 2034 | Fall 18 | A |
| 1945 | 1263 | Fall 19 | B+ |
| 2654 | 2015 | Fall 19 | B |

This value is not allowed because this value is not defined as a primary key in the course table.

► Enterprise Constraints



- ▶ Sometimes referred to as semantic constraints are additional rules specified by users or database administrators and can be based on multiple tables.



► Business Rules



Business rules are obtained from users when gathering requirements.

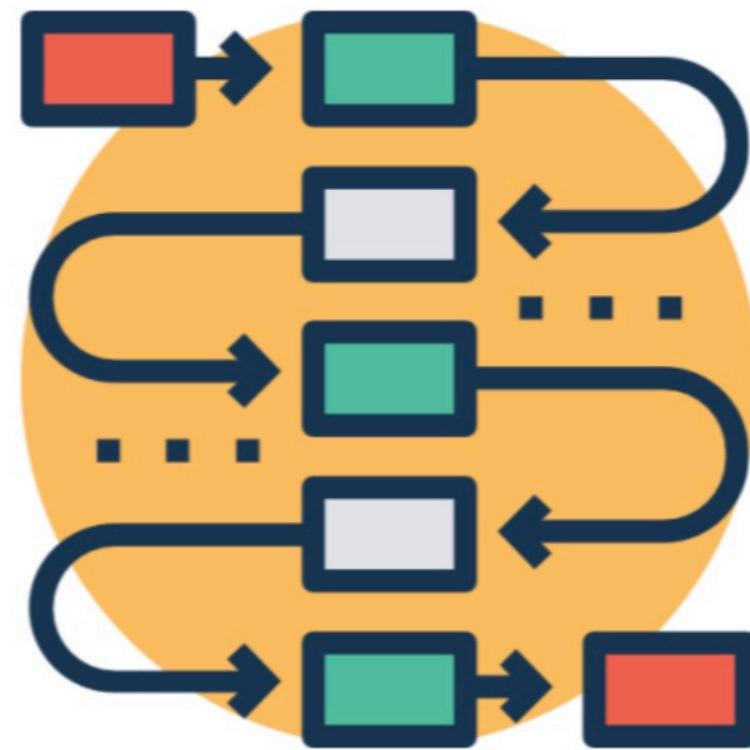
Some examples of business rules are:

- ▶ A teacher can teach many students.
- ▶ A class can have a maximum of 35 students.
- ▶ A course can be taught many times, but by only one instructor.
- ▶ Not all teachers teach classes.



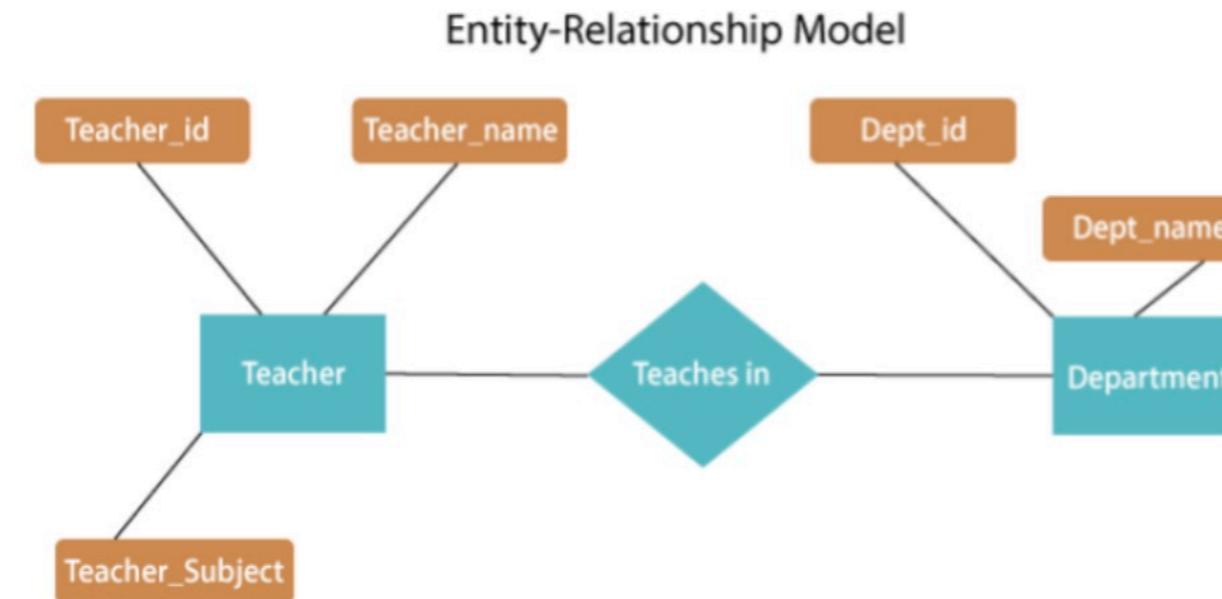


Entity Relationship Diagram



► Entity-Relationship Model

- ❖ An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge.
- ❖ A basic ER model is composed of **entity types** (which classify the things of interest) and specifies **relationships** that can exist between entities (instances of those entity types).



►Entity



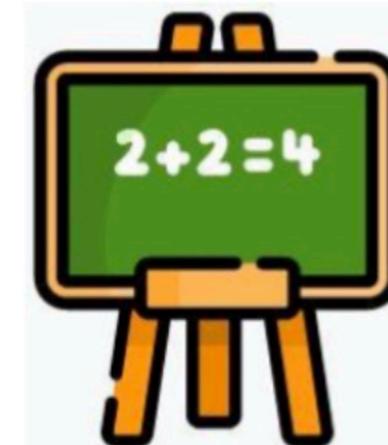
An entity is **an object** in the real world with an independent existence that can be differentiated from other objects.

An entity might be:

- **physical existence** (e.g., a lecturer, a student, a car)
- **conceptual existence** (e.g., a course, a job, a position)



a student

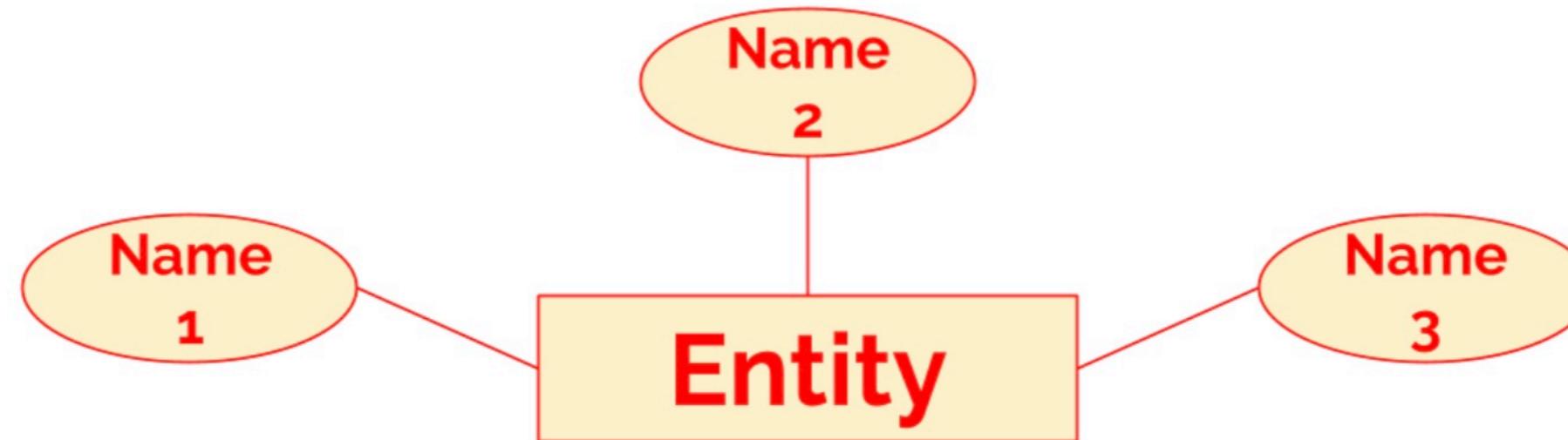


a course

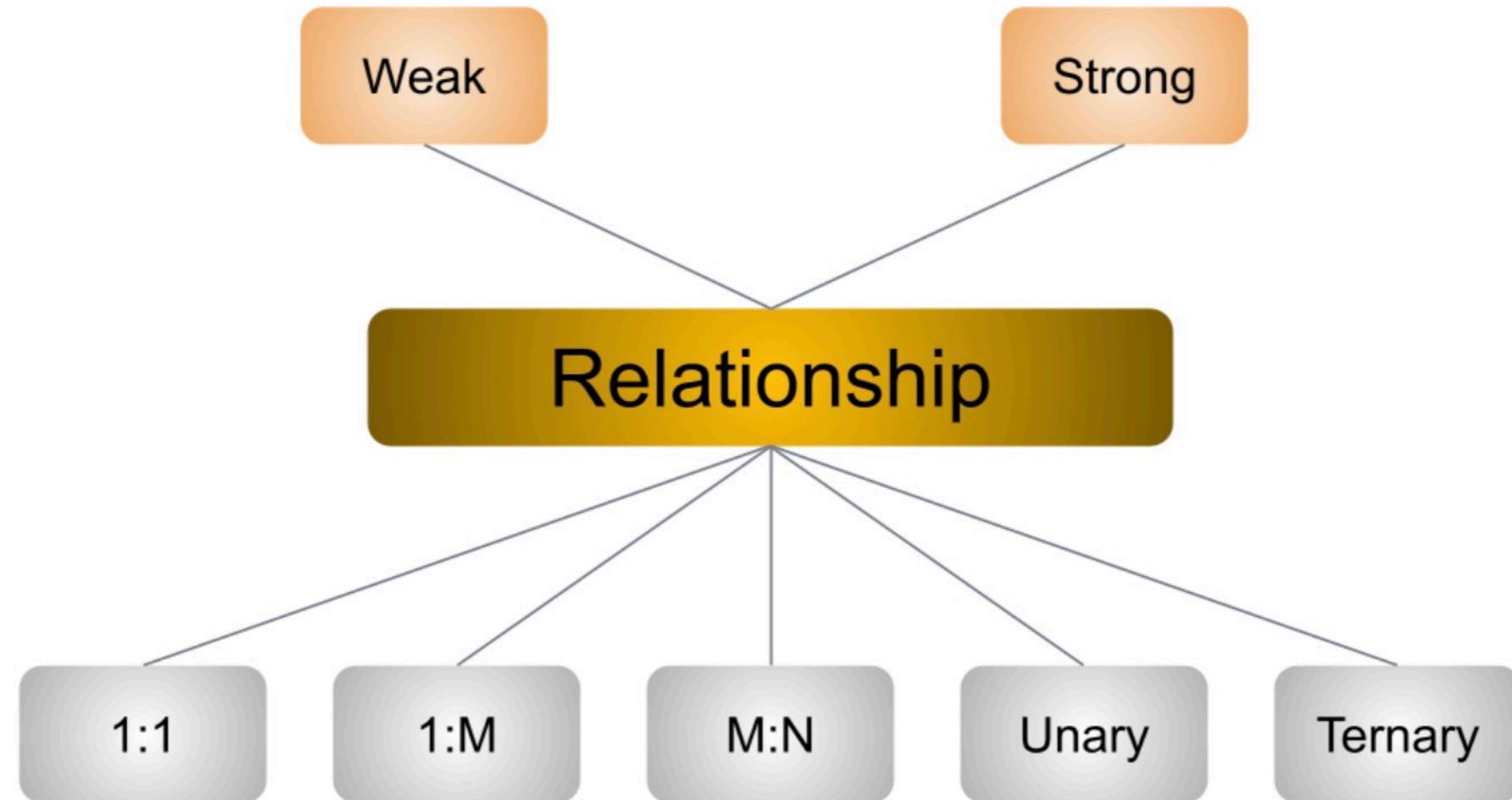


►Attributes

- Each entity is described by a set of attributes.
- Each attribute has **a name and is associated with an entity and a domain of legal values.**
- However, the information about attribute domain is not presented on the ERD.



►Relationships

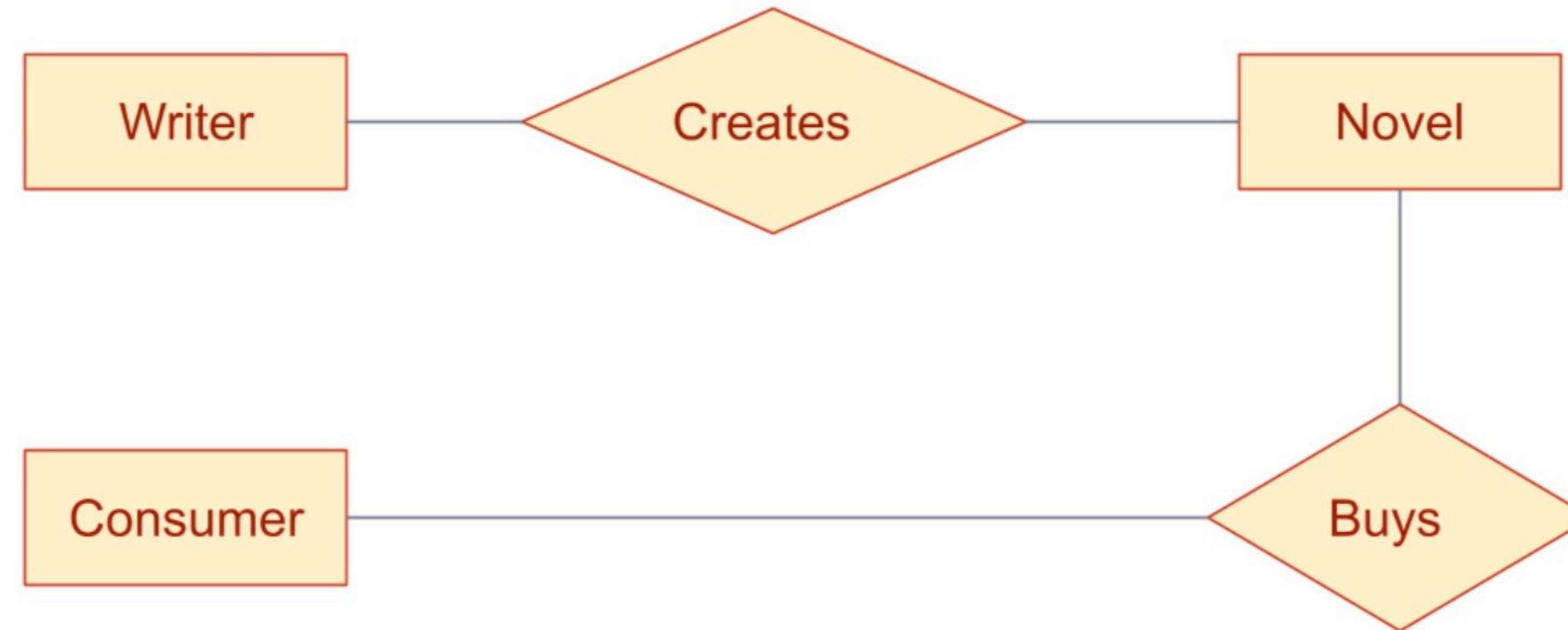


►What is ER diagram?



An Entity Relationship Diagram (ERD) is a **visual representation of different entities** within a system and how they **relate to each other**.

For example, the elements **writer**, **novel**, and a **consumer** may be described using ER diagrams the following way:

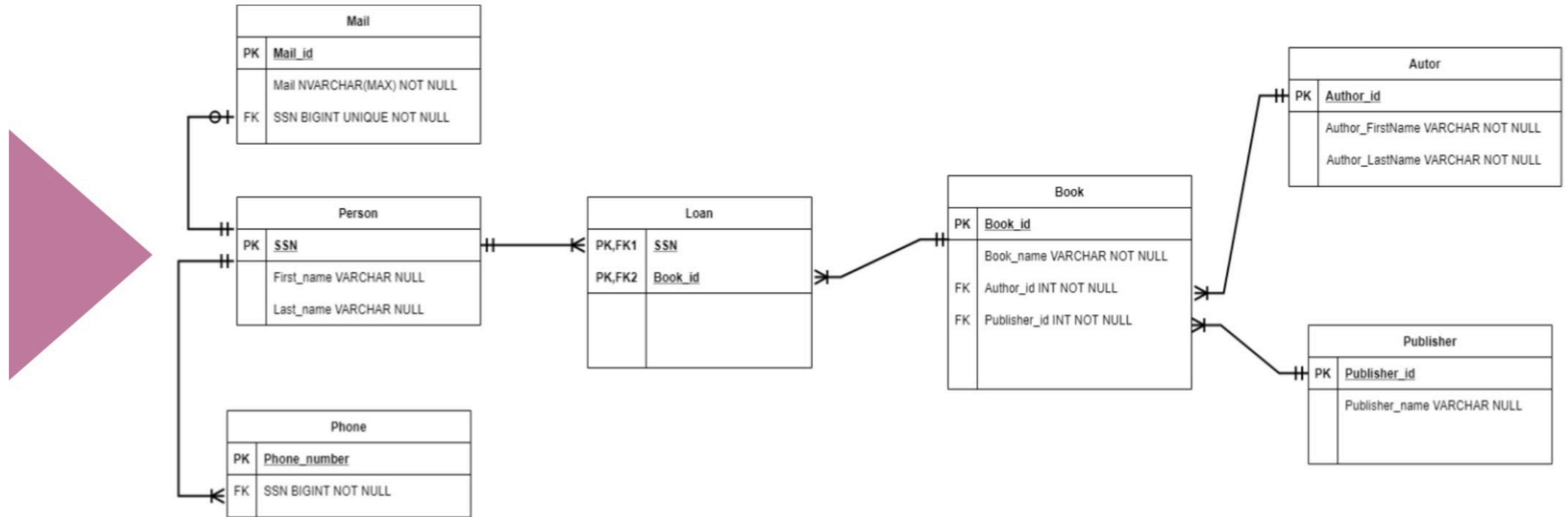


►ERD Notations



The two of notations most widely used for creating Entity Relationship Diagrams are:

- ▶ Chen's notation
- ▶ Crow's foot notation



►How to Draw ER Diagrams

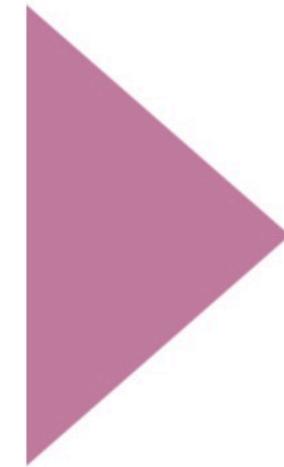


- ❖ **Identify all the entities** in the system. An entity should appear only once in a particular diagram.
- ❖ **Identify relationships** between entities. Connect them using a line and add a diamond in the middle describing the relationship.
- ❖ **Add attributes** for entities. Give meaningful attribute names so they can be understood easily.

► ER Diagram Best Practices



- ❖ Provide a **precise and appropriate name** for each entity, attribute, and relationship in the diagram. In naming entities, remember to use singular nouns.
- ❖ **Remove** vague, redundant or unnecessary relationships between entities.
- ❖ **Never connect** a relationship to another relationship.
- ❖ Make effective use of colors. You can use colors to classify similar entities or to highlight key areas in your diagrams.



Let's look up Diagram of the Sample Retail DB



Keys Constraints



► Keys



- ❖ Keys are very important part of Relational data model. They are used to **establish and identify relationships between tables** and also to **uniquely identify any record or row** of data inside a table.
- ❖ A Key can be a single attribute or a group of attributes, where the combination may act as a key.
- ❖ There are several types of keys, such as **Candidate key**, **Composite key**, **Primary key (PK)**, **Alternate key** and **Foreign key (FK)**.



►Types of Keys - 1

- ❖ **Candidate key:** A candidate key is a simple or composite key that is **unique and minimal**.
- ❖ **Composite key:** A composite key is composed of **two or more attributes**.
- ❖ **Primary key:** The primary key (PK) is a candidate key that is selected by the database designer to be used as an identifying mechanism for the whole entity set. It must **uniquely identify tuples in a table and not be null**. The primary key is indicated in the ER model by **underlining the attribute**.



►Types of Keys - 2

- ◆ **Alternate key:** Alternate keys are all candidate keys not chosen as the primary key.
- ◆ **Foreign key:** A foreign key (FK) is an attribute in a table that references **the primary key in another table**. Both foreign and primary keys must be of the same data type.



► PRIMARY KEY



A primary key is a column or a group of columns that uniquely identifies each row in a table. You create a primary key for a table by using the **PRIMARY KEY** constraint.



- ❖ In case the **primary key has two or more columns**, you must use the PRIMARY KEY constraint as a **table constraint**.
- ❖ Each table can contain **only one primary key**.
- ❖ SQL Server automatically sets the **NOT NULL** constraint for all the primary key columns.
- ❖ SQL Server also automatically **creates a unique clustered index** (or a non-clustered index if specified as such) when you create a primary key.

► FOREIGN KEY

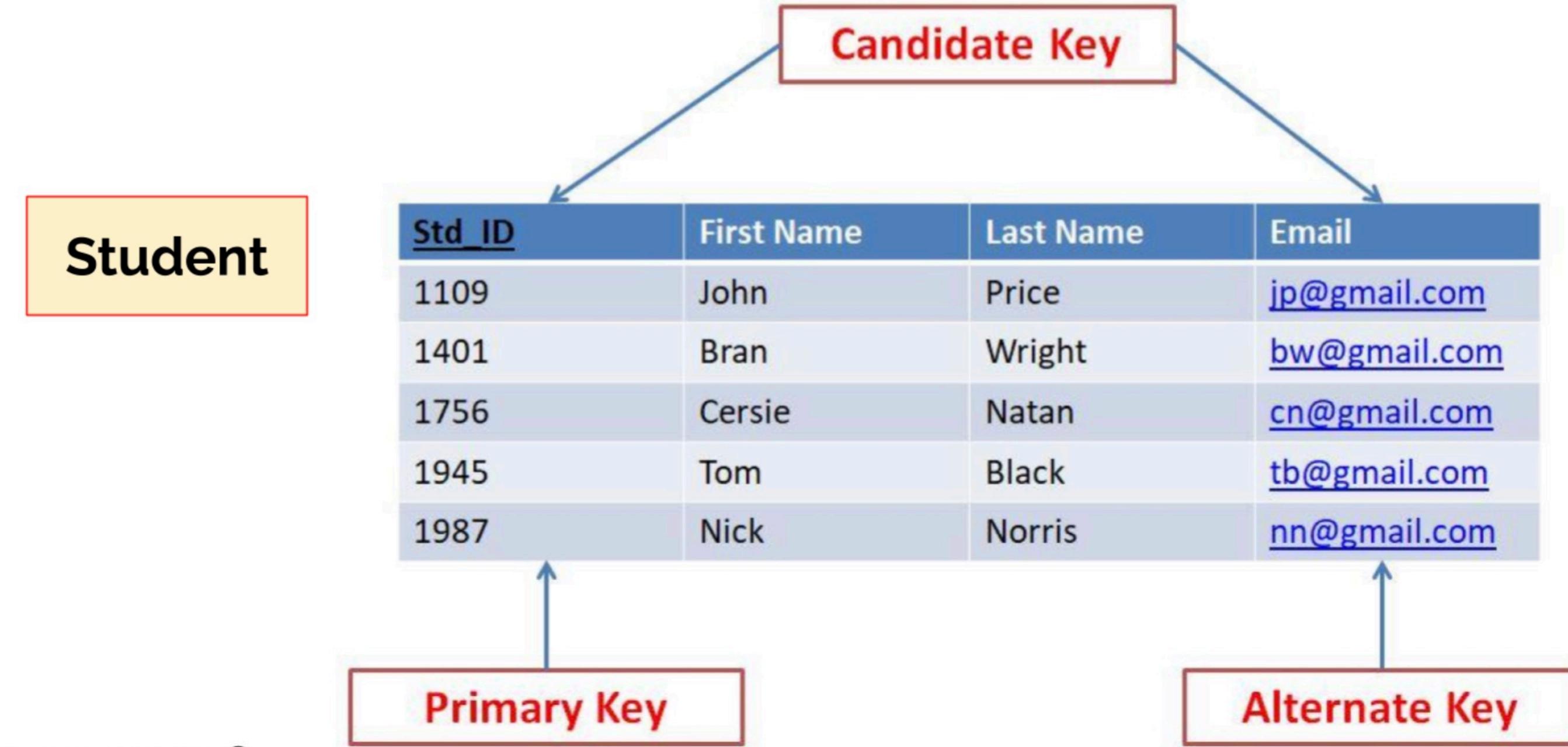


The **FOREIGN KEY** (FK) constraint defines a column, or combination of columns, whose values match the **PRIMARY KEY** (PK) of another table.

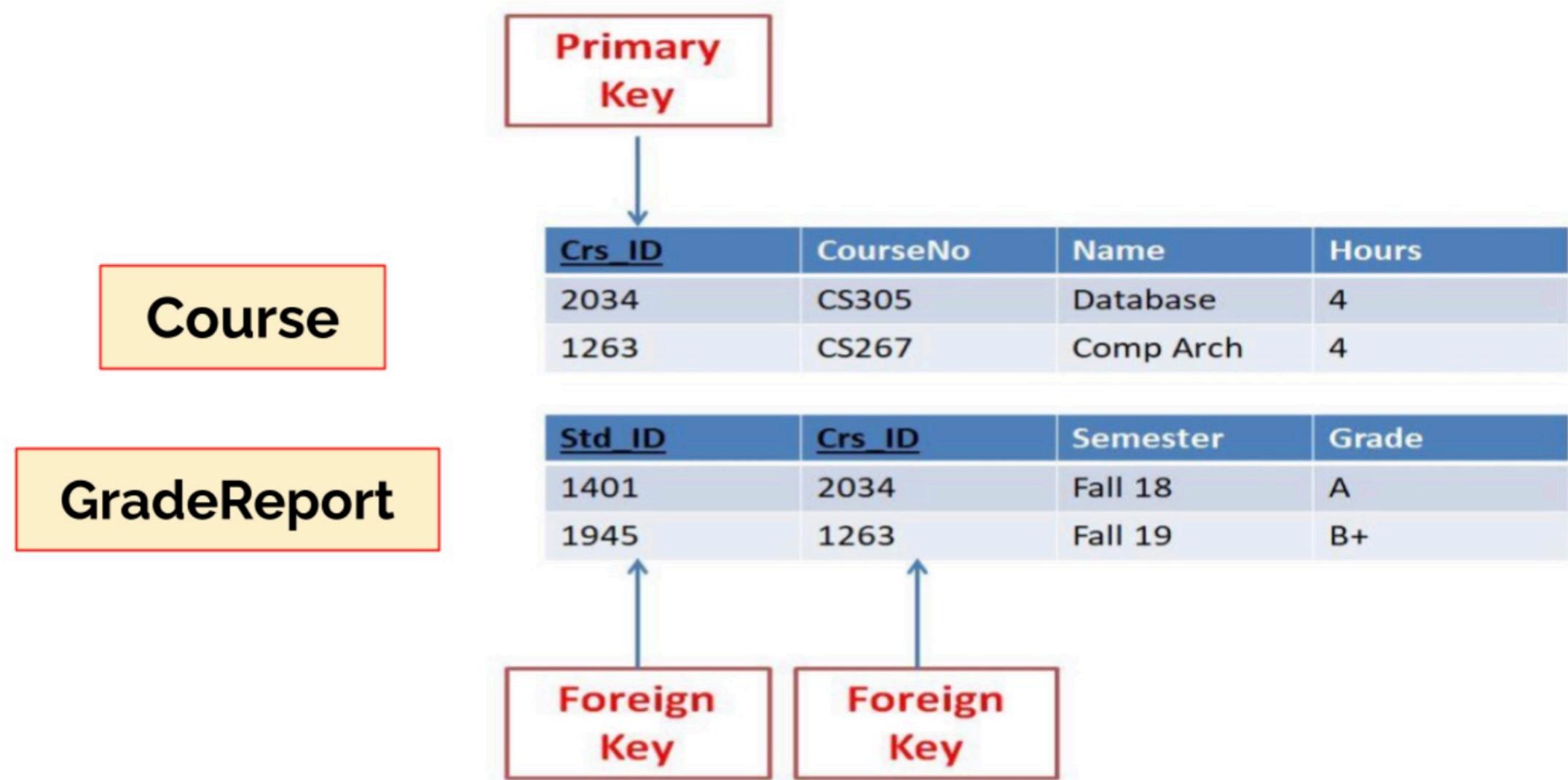


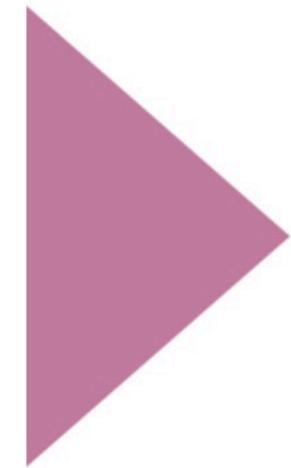
- ❖ Values in an **FK** are automatically updated when the **PK** values in the associated table are updated/changed.
- ❖ **FK** constraints must reference **PK** or the **UNIQUE** constraint of another table.

► Example of Keys - 1



►Example of Keys - 2





Let's look up Sample Retail DB



Normalization



► What Is Normalization?



- ❖ Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like **Insertion, Update and Deletion Anomalies**.

- ❖ Normalization rules divides larger tables into smaller tables and links them using relationships.

► Why Normalization?



- ❖ Avoid Anomalies
- ❖ Speed
- ❖ Reduce Data Redundancy

► What are the Anomalies



- ❖ Deletion Anomaly
- ❖ Insertion Anomaly
- ❖ Update Anomaly

| Personal_ID | Name | Lastname | Fullname | Phone | E_mail | Book | Author | Publisher |
|-------------|-------|----------|--------------|-----------------------|--|------------------------------------|-------------------------|------------------|
| 1 | Bob | Marley | Bob Marley | 5056659595/5336559863 | bobmarley@gmail.com | Crime And Punishment - Dostoyevski | | Collins |
| 1 | Bob | Marley | Bob Marley | 5056659595/5336559863 | bobmarley@gmail.com | Dead Souls | Gogol | Collins |
| 1 | Bob | Marley | Bob Marley | 5056659595/5336559863 | bobmarley@gmail.com | The Hunchback of Notre-Dame | Victor Hugo | Hachette |
| 2 | Bart | Simpson | Bart Simpson | 5078893526 | bart@gmail.com | Mother / Of Mice and Men | Maxim Gorki / Steinbeck | Hachette / Simon |
| 3 | Lisa | Marley | Lisa Marley | 5532558963 | | Les Miserables | Victor Hugo | Hachette |
| 4 | Bruce | Lee | Bruce Lee | 5325558963/5445632586 | lee@gmail.com | Dead Souls | Gogol | Simon |
| 4 | Bruce | Lee | Bruce Lee | 5325558963/5445632586 | lee@gmail.com | The Hunchback of Notre-Dame | Victor Hugo | Hachette |

► How to Avoid Anomalies



- ★ The best approach to creating tables without anomalies is to ensure that **the tables are normalized**, and that's accomplished by understanding **functional dependencies**.
- ★ FD ensures that all attributes in a table belong to that table. In other words, it will **eliminate redundancies** and **anomalies**.

► Normal Forms



All the tables in any database can be in one of the normal forms.

Ideally we only want **minimal redundancy** for PK to FK.

There are six normal forms, but we will only look at the first four, which are:

- ▶ First normal form (1NF)
- ▶ Second normal form (2NF)
- ▶ Third normal form (3NF)
- ▶ ...

► First Normal Form (1NF)



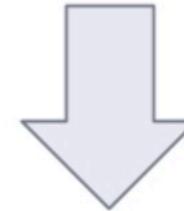
- There must be a **primary key** in each entity table.
- A table must not contain more than one domain.
(atomic)
- Each column should hold only one information.
- There should be no repeating columns.

| Primary key | | | |
|-------------|------------|-------|----------|
| Personal_ID | Phone | Name | Lastname |
| 1 | 5056659595 | Bob | Marley |
| 1 | 5336559863 | Bob | Marley |
| 2 | 5078893526 | Bart | Simpson |
| 3 | 5532558963 | Lisa | Marley |
| 4 | 5325558963 | Bruce | Lee |
| 4 | 5445632586 | Bruce | Lee |

► Second Normal Form (2NF)

- For the second normal form, the relation must first be in 1NF.
- If, and only if, the PK comprises a single attribute the relation is automatically in 2NF.
- If the relation has a **composite PK**, then each non-key attribute must be **fully dependent** on the entire PK and not on a subset of the PK (i.e., there must be no **partial dependency** or **augmentation**).

| primary key | | | |
|-------------|---------|-----------------------------|--------------------------|
| Personal_ID | Book_ID | Book | Author |
| | 1 | 10 | Fyodor Dostoyevski |
| | 1 | 20 | Nikolay Vasilyeviç Gogol |
| 2 | 30 | Mother | Maxim Gorki |
| 2 | 40 | Of Mice and Man | John Steinbeck |
| 3 | 50 | Les Misérables | Victor Hugo |
| 1 | 60 | The Hunchback of Notre-Dame | Victor Hugo |
| 4 | 70 | Dead Souls | Nikolay Vasilyeviç Gogol |
| 4 | 60 | The Hunchback of Notre-Dame | Victor Hugo |



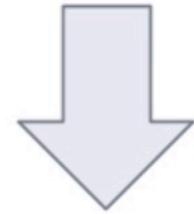
| primary key | |
|-------------|---------|
| Personal_ID | Book_ID |
| | 10 |
| 1 | 20 |
| 2 | 30 |
| 2 | 40 |
| 3 | 50 |
| 1 | 60 |
| 4 | 70 |
| 4 | 60 |

► Third Normal Form (3NF)

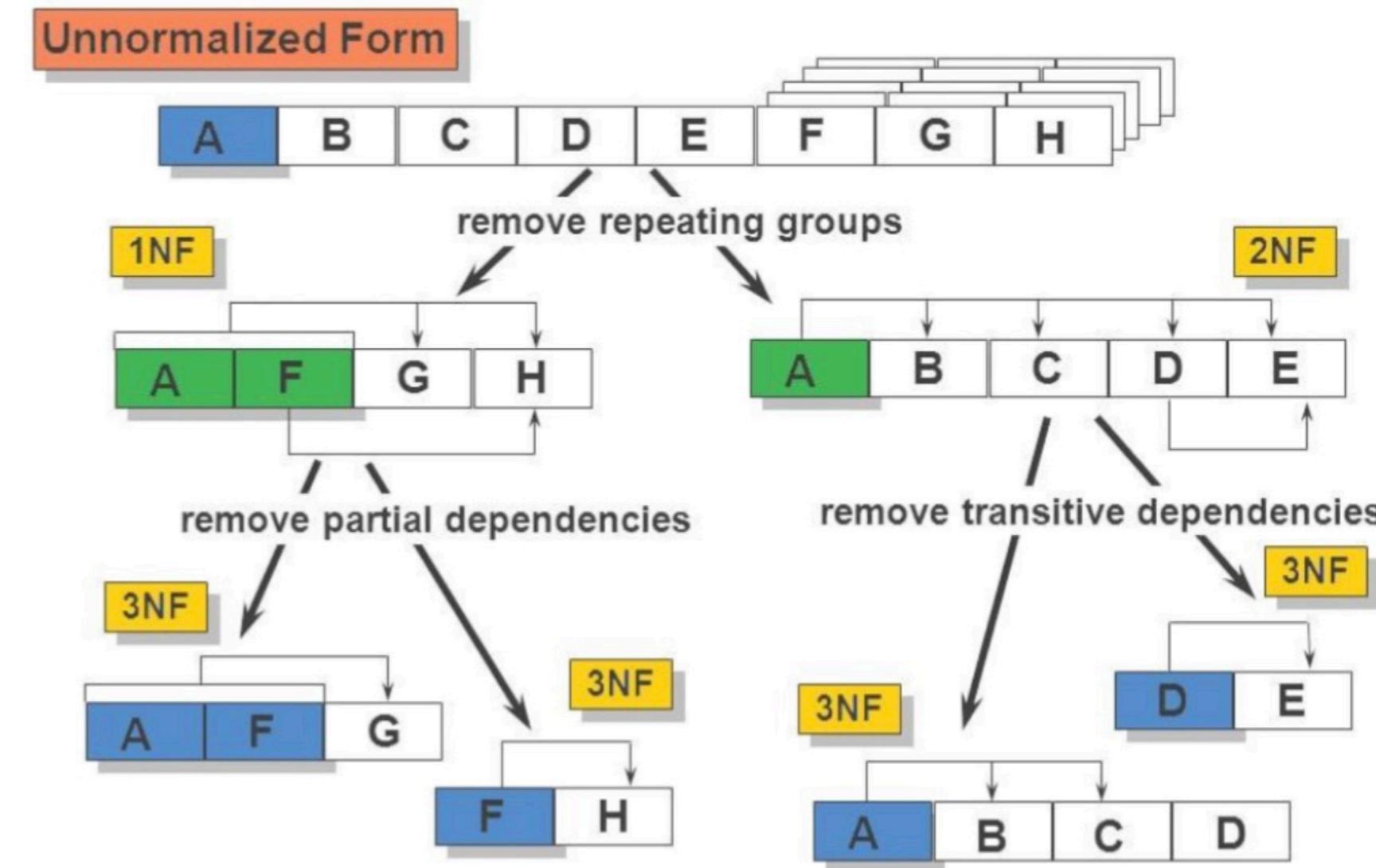


- To be in third normal form, the relation must be in second normal form.
- Also all **transitive dependencies** must be removed; a **non-key attribute** may not be functionally dependent on another non-key attribute.

| Primary key | | | |
|-------------|------------|-------|----------|
| Personal_ID | Phone | Name | Lastname |
| 1 | 5056659595 | Bob | Marley |
| 1 | 5336559863 | Bob | Marley |
| 2 | 5078893526 | Bart | Simpson |
| 3 | 5532558963 | Lisa | Marley |
| 4 | 5325558963 | Bruce | Lee |
| 4 | 5445632586 | Bruce | Lee |



| Primary key | |
|-------------|------------|
| ID | Phone |
| 1 | 5056659595 |
| 1 | 5336559863 |
| 2 | 5078893526 |
| 3 | 5532558963 |
| 3 | 5325558963 |
| 4 | 5445632586 |



► Normalize Library Database Tables

| Personal_ID | Name | Surname |
|-------------|-------|---------|
| 1 | Bob | Marley |
| 2 | Bart | Simpson |
| 3 | Lisa | Marley |
| 4 | Bruce | Lee |

| Book_ID | Book | Author_ID | Publisher_id |
|---------|-----------------------------|-----------|--------------|
| 10 | Crime And Punishment | 1 | 1 |
| 20 | Dead Souls | 2 | 1 |
| 30 | Mother | 3 | 2 |
| 40 | Of Mice and Man | 4 | 3 |
| 50 | Les Miserables | 5 | 2 |
| 60 | The Hunchback of Notre-Dame | 5 | 2 |
| 70 | Dead Souls | 2 | 3 |

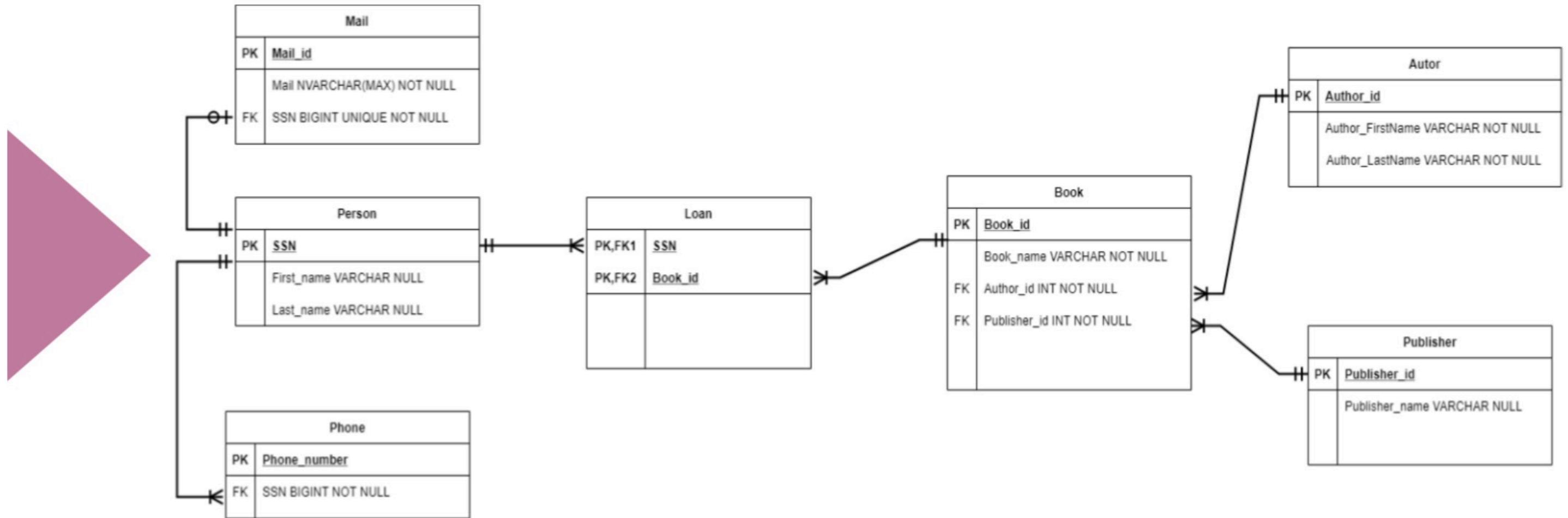
| E_mail_ID | E_mail | Personal_ID |
|-----------|--|-------------|
| 1 | bobmarley@gmail.com | 1 |
| 2 | bart@gmail.com | 2 |
| 3 | lee@gmail.com | 4 |

| Personal_ID | Book_ID |
|-------------|---------|
| 1 | 10 |
| 1 | 20 |
| 2 | 30 |
| 2 | 40 |
| 3 | 50 |
| 1 | 60 |
| 4 | 70 |
| 4 | 60 |

| Author_ID | Name |
|-----------|--------------------------|
| 1 | Fyodor Dostoyevski |
| 2 | Nikolay Vasilyeviç Gogol |
| 3 | Maxim Gorki |
| 4 | John Steinbeck |
| 5 | Victor Hugo |

| Personal_ID | Phone |
|-------------|------------|
| 1 | 5056659595 |
| 1 | 5336559863 |
| 2 | 5078893526 |
| 3 | 5532558963 |
| 3 | 5325558963 |
| 4 | 5445632586 |

| Publisher_ID | Publisher |
|--------------|-----------|
| 100 | Collins |
| 200 | Hachette |
| 300 | Simons |



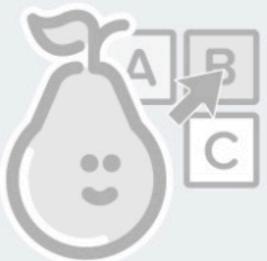
Is everything clear so far?



CJS Students choose an option

REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar



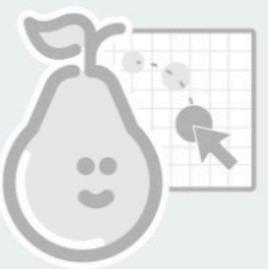
No Multiple Choice Response
You didn't answer this question

How well did you like this lesson?



Students, drag the icon!

Pear Deck Interactive Slide
Do not remove this bar



No Draggable™ Response
You didn't answer this question