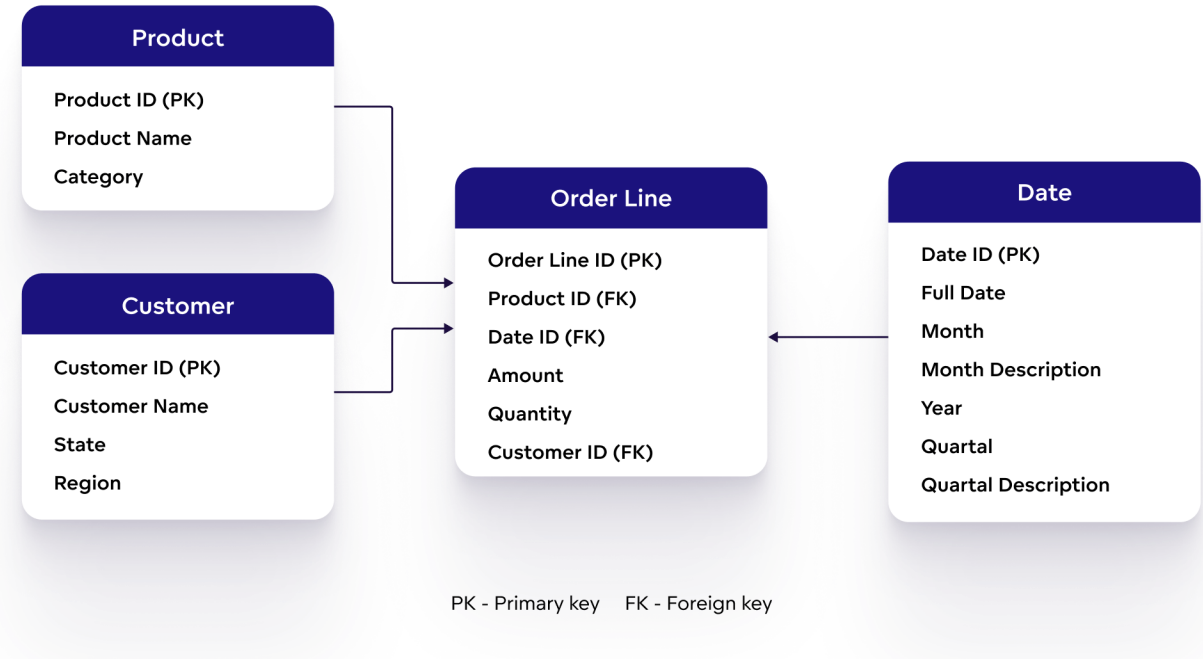


# The relational model

# Data Model

- A *data model* is representation of *data elements* and the relationships between them based on real-world objects
  - For example when modeling a simple online store, data element representing customer is related to a data element representing an order
- Data elements document real-world which means that the data model represents reality



# Components of a data model

- A data model consists of three components:
  - i. *Structural part*: a set of rules according to which databases can be constructed
  - ii. *Integrity part*: a set of integrity constraints to ensure database integrity
  - iii. *Manipulative part*: a set of operations that are allowed on the data

# The relational model

- When all data model's data is logically structured within *relations*, the model is a *relational model*
- These relations are informally referred to as *tables*
- The data is perceived by the users as tables
- Relation has named *attributes* (informally called *columns*)
- Attributes have a set of allowable values, which is referred to as the attribute's *domain*
  - For example "person" relation's "age" attribute could be an integer value larger or equal to zero
- The actual data is in relations's *tuples* (informally called *rows*)

The diagram illustrates a database table structure with the following components and annotations:

- relation name:** An arrow points from the label "relation name" to the word **COURSE**.
- attributes:** An arrow points from the label "attributes" to the header row of the table.
- attribute names:** An arrow points from the label "attribute names" to the **code** header.
- tuples:** Three arrows point from the label "tuples" to the three data rows of the table.

<b>code</b>	<b>name</b>	<b>credits</b>
swd1tf001	Introduction to Software Engineering	5
swd4tf002	Programming (Java)	5
swd4tf003	Data Management and Databases	5

# Properties of relations

- Each relation has a name that is distinct from all other relation names
- Each attribute has a distinct name
- Each tuple's cell contains exactly one value
- Values of an attribute are all from the same domain
- The order of attributes has no significance
- There are no duplicate tuples
- The order of tuples has no significance

# Integrity constraints

- The quality of the data directly determines the quality of the whole database
- Therefore preventing entry of incorrect data is one of the most important functions of a DBMS
- *Integrity constraints* are used to control the legal database states
- If the database satisfies all the integrity constraints specified on the database schema, it is in a legal state

# Domain integrity

- A *domain constraint* specifies the set of allowable values for a column
  - For example valid grade marks are integers between 0 and 5
- Domain constraints enforce *domain integrity*



# Entity Integrity

- A *superkey* is an attribute or group of attributes that uniquely identifies each tuple of a relation
- Relation can have multiple superkeys
  - In the "course" relation the "code" attribute, and group of "code" and "name" attributes are superkeys
  - What other superkeys does the "course" relation have?

code	name	credits
swd1tf001	Introduction to Software Engineering	5
swd4tf002	Programming (Java)	5
swd4tf003	Data Management and Databases	5

# Entity Integrity

- A *candidate key* is a superkey that satisfies the property of *minimality*
  - Minimality is satisfied if an attribute can't be removed from the group of attributes without breaking the uniqueness property
  - In the "course" relation the group of "code" and "name" attributes doesn't satisfy minimality, so it isn't a candidate key
  - What other candidate keys does the "course" relation have?
- From the set of candidate keys for the relation, *exactly one* candidate key is chosen to be the *primary key*
- The other candidate keys become *alternate keys*
- Each tuple has a value for the primary key, it can't be missing
- *Primary key constraint* prevents duplicate tuples to exist for the relation
- Primary key constraints enforce *entity integrity*

# Referential Integrity

- *Foreign key* is a attribute or group attributes whose values are required to match those of the primary key of the referenced relation
- There can be several foreign keys in a relation
- Foreign-to-primary-key matching is the "glue" which holds the database together
- *Foreign key constraint* prevents foreign key not being matched by a primary key in the referenced relation
- Foreign key constraints enforce *referential integrity*

Primary key  
constraint

empno	empname	deptno
20	Mark	10
18	Sue	10
49	Frank	20
31	Mary	20

Foreign key  
constraint

Primary key  
constraint

deptno	deptname	budget
10	Sales	50000
20	Marketing	250000

**Employee** (Referencing relation)

**Department** (Referenced relation)