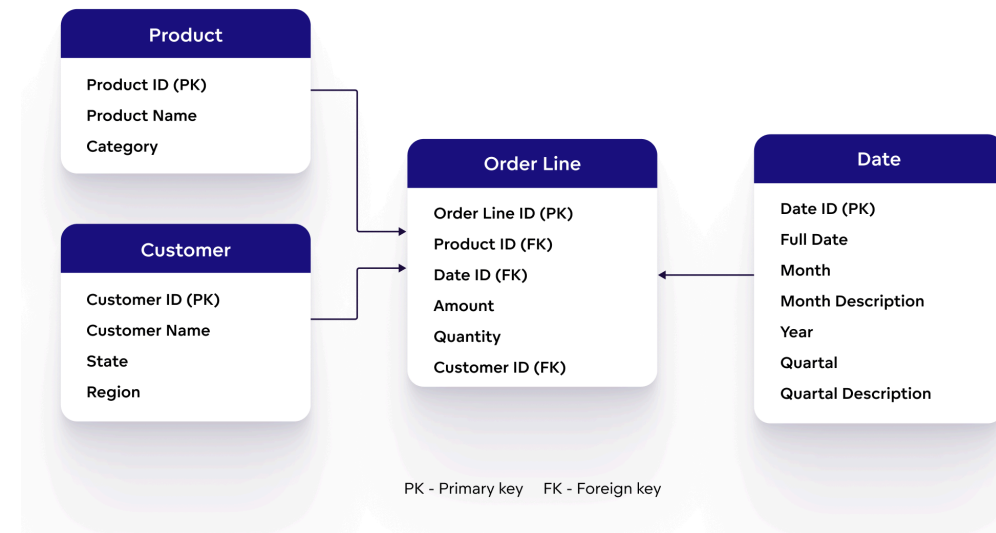


The relational model

- The learning objectives for this week are:
 - Knowing what is a *data model*
 - Knowing what is the *relational data model*
 - Knowing the core terminology of the relational model
 - Knowing what are the properties of database *relations*
 - Knowing what are *domain integrity*, *entity integrity*, and *referential integrity* rules
 - Knowing how to identify *candidate keys*, *primary keys*, *alternate keys*, and *foreign keys*

Data Model

- A *data model* is an abstract representation of *data elements* and the relationships between them based on real-world objects
 - In case of a simple online store, data element representing customer consists of data elements such as customer's name and region
- 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 cell.
- tuples:** Three arrows point from the label "tuples" to the three data rows of the table.

| code | name | credits |
|-------------|--------------------------------------|----------------|
| 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 of a relation 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 rules 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 an attribute
 - 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?

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

Entity Integrity

- 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's value *should not change*
 - For example person's name or phone number might sound tempting options for a primary key but are actually subject to change
- *Primary key constraint* prevents duplicate tuples to exist for the relation
- Primary key constraints enforce *entity integrity*

Surrogate keys

- If there is initially no candidate key for a relation, then we cannot determine a *natural* primary key
- For example a relation for email messages:

| from | to | title | body |
|----------------------------|----------------------------|----------|--------------|
| kalle.ilves@haaga-helia.fi | john.doe@gmail.com | Greeting | Hello John! |
| john.doe@gmail.com | kalle.ilves@haaga-helia.fi | Response | Hello Kalle! |

Surrogate keys

- We have to take care of the situation by including an extra attribute in the relation to act as the primary key
- For example a "messageid" column that holds a unique number for each tuple:

| messageid | from | to | title | body |
|-----------|----------------------------|----------------------------|----------|--------------|
| 1 | kalle.ilves@haaga-helia.fi | john.doe@gmail.com | Greeting | Hello John! |
| 2 | john.doe@gmail.com | kalle.ilves@haaga-helia.fi | Response | Hello Kalle! |

Surrogate keys

- Such primary key is called a *surrogate key*
- Surrogate key has no relationship to the real-world meaning of the data held in a tuple
- Surrogate keys are quite common and a natural key is often replaced with a surrogate key
- Surrogate keys are commonly generated by the DBMS once a tuple is inserted
- Automatically incremented numbers and randomly generated values like UUID are common surrogate key values

Choosing a primary key

- Let's consider a suitable primary key in the following cases:
 - Is "course" relation's "name" attribute a good option for a primary key? Why or why not?
 - Person's contacts are stored in relation which has attributes "address" and "name" (home address and name of a contact). What would be suitable primary key for this relation and why?

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)

Example of primary and foreign keys

- Let's have a look at the exercise 5 in the first week's intro assignment
- *What are the primary keys for each table?*
- *What are the foreign keys for each table?*

TEAM

| teamno | team name |
|--------|-----------|
| 9 | Hawks |
| 7 | Tigers |
| 5 | Sharks |

ARTIST

| artistno | given name | family name |
|----------|------------|-------------|
| a15 | Katy | Perry |
| a3 | Ariana | Grande |
| a16 | Bruno | Mars |
| a20 | Johnny | Smith |
| a7 | Lady | Gaga |
| a12 | Alicia | Keys |

TEAM_ARTIST

| teamno | artistno |
|--------|----------|
| 9 | a3 |
| 7 | a7 |
| 7 | a16 |
| 9 | a7 |
| 7 | a12 |

Not null constraint

- *Null* is a marker for a missing attribute value
- Null is not the same as e.g. blanks or zero. Null represents absence of a value
- The *not null constraint* is a restriction placed on an attribute
- It enforces the condition that, in that attribute, every tuple of data must contain a value
- For example it would make sense that in the "employee" relation, the "deptno" attribute has a not null constraint, meaning that every employee belongs to a department

Database manipulation

- A *manipulation mechanism* is among the most important parts of a data model
- A manipulation mechanism allows the data to be retrieved and updated
- SQL is the standard database language for relational databases. With SQL we can:
 - Create the database and relation structures
 - Perform insertion, modification, and deletion of data from the relations
 - Perform database queries
- Instead of using formal terms of relations, attributes, and tuples, the terms *tables*, *columns*, and *rows* are used in the SQL standard

SQL

- An SQL query is a single statement in which you describe what you want from the database
- The query operates on tables and builds a result table from one or more tables in the database
- Here's an example of an SQL query:

```
SELECT code, name, credits  
FROM course  
WHERE name = 'Data Management and Databases';
```

Summary

- A *data model* consists of three components: the *structural part*, the *integrity part* and the *manipulative part*
- In the *relational model*, all data is logically structured within relations that have attributes and tuples
- *SQL* is the standard database language for relational databases
- *Integrity constraints* are rules which make sure that the database is in a legal state
- *Domain constraint* specifies the set of allowable values for an attribute
- *Primary key constraint* prevents duplicate tuples to exist for the relation
- *Foreign key constraint* prevents foreign key not being matched by a primary key in the referenced relation