

**PROJECT**

KUC0396

Date: 2nd June 2024 Adam Kuchár

1. **DD S01 L02**

**Vysvětlete rozdíl mezi pojmem data a informace – příklad ve vašem projektu psáno anglicky**

**Data**: Data are raw, unprocessed facts that are collected and stored. They represent individual values or pieces of information that are not meaningful by themselves. In the context of a database, data are the individual entries stored in tables without context or interpretation.  
  
**Information**: Information is the result of processing data to give it context and meaning, allowing us to make informed decisions

**Represenation of Data**: In my database, columns material\_name, material\_description, material\_quantity, and material\_image represent raw data which are stored in table materials.  
  
**Representation of Information:** For instance, if we want to understand the total quantity of materials available we can sum up the material\_quantity values, giving us an insight into the total materials available.

# **DD S02 L02**

**Entity, instance, atributy a identifikátory – popište v příkladech na vašem projektu**

An entity represents a real-world object or concept in the database. Each entity corresponds to a table in the database.  
  
Example Entites in my project:

1. Person
2. Client
3. Task
4. Materials

**Instance** – is a single occurence or record of an entity, in a database table, each row represents an instance.

Instance for the **person** entity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_person | first\_name | last\_name | email | phone\_number |
| 1 | Adam | Kuchar | adam@gmail.com | 1223456789 |

Instance for the **client** entity

|  |  |
| --- | --- |
| id\_client | company\_name |
| 1 | Kuchár Inc. |

**Attributes** are the properties or details of an entity. In a table, attributes are represented by columns.

**Example Attributes:**

Person entity – id\_person, first\_name, last\_name, email, phone\_number

Materials entity – id\_materials, metarial\_name, material\_description, material\_quantity, material\_iamge

**Identifiers** are unique attributes used to distinguish each instance of an entity. The primary key is the main identifier for an entity.

**Example Identifiers:**

For the **person** entity: id\_person is the primary key

For the **materials** entity: id\_materials is the primary key

# **DD S03 L01**

**Popište všechny vztahy ve vaší databázi v angličtině, včetně kardinality a povinnosti členství – každý vztah dvěma větami**

1. **Address and Place of Work (place\_of\_work\_address\_fk):**

Each place of work has one address, with id\_address in the place\_of\_work table referencing id\_address in the address table. This is a mandatory one-to-many relationship, meaning each place of work must have an address, but an address can be associated with multiple places of work.

1. **Certification and Profession (certification\_profession\_fk):**

Each certification is associated with one profession, indicated by id\_profession in the certification table referencing id\_profession in the profession table. This is a mandatory one-to-many relationship where each profession can have multiple certifications, but each certification belongs to only one profession.

1. **Client and Person (client\_fk):**

Each client corresponds to one person, with id\_client in the client table referencing id\_person in the person table. This is a mandatory one-to-one relationship where every client must be linked to a person.

1. **Client Task and Client (client\_task\_client\_fk):**

Each entry in the client\_task table links one client to one task, with id\_client referencing id\_client in the client table. This is a mandatory many-to-one relationship, meaning each client task entry must have a client, and a client can have multiple task entries.

1. **Client Task and Task (client\_task\_task\_fk):**

Each entry in the client\_task table also links to one task, with id\_task referencing id\_task in the task table. This is a mandatory many-to-one relationship where each task can have multiple client task entries, but each client task entry is associated with one task.

1. **Feedback and Client/User (feedback\_client\_user\_chk, feedback\_client\_fk, feedback\_user\_fk):**

Each feedback entry can be linked to either a client or a user but not both simultaneously, enforced by a check constraint. This is an optional one-to-many relationship where clients and users can have multiple feedback entries, but each feedback entry is linked to either one client or one user

1. **Materials Used and Materials (materials\_used\_materials\_fk):**

Each materials used entry links to one material, with id\_materials referencing id\_materials in the materials table. This is a mandatory many-to-one relationship where each material can be used in multiple work records, but each entry in the materials\_used table must reference one material.

1. **Materials Used and Work Record (materials\_used\_work\_record\_fk):**

Each materials used entry also links to one work record, with id\_work\_record referencing id\_work\_record in the work\_record table. This is a mandatory many-to-one relationship where each work record can use multiple materials, but each entry in the materials\_used table must reference one work record

1. **Person and User (userr\_fk):**

Each user corresponds to one person, with id\_user in the userr table referencing id\_person in the person table. This is a mandatory one-to-one relationship where every user must be linked to a person.

1. **Place of Work and Work Record (work\_record\_place\_of\_work\_fk):**

Each work record is associated with one place of work, with id\_place\_of\_work in the work\_record table referencing id\_place\_of\_work in the place\_of\_work table. This is a mandatory many-to-one relationship where each place of work can have multiple work records, but each work record must reference one place of work.

1. **Profession and User Profession (user\_profession\_profession\_fk):**

Each entry in the user\_profession table links one user to one profession, with id\_profession referencing id\_profession in the profession table. This is a mandatory many-to-one relationship where each profession can be linked to multiple users, but each user profession entry must reference one profession.

1. **Task and Client (task\_client\_fk):**

Each task is linked to one client, with id\_client in the task table referencing id\_client in the client table. This is a mandatory many-to-one relationship where each client can have multiple tasks, but each task must reference one client.

1. **Task and Task (Self-Referencing) (task\_parent\_fk):**

Each task can have a parent task, with parent\_task\_id in the task table referencing id\_task in the same table. This is an optional many-to-one relationship where tasks can be hierarchically organized.

1. **Task and User (task\_userr\_fk):**

Each task is assigned to one user, with id\_user in the task table referencing id\_user in the userr table. This is a mandatory many-to-one relationship where each user can have multiple tasks, but each task must reference one user.

1. **User and Profession (User Profession) (user\_profession\_userr\_fk):**

Each entry in the user\_profession table links one user to one profession, with id\_user referencing id\_user in the userr table. This is a mandatory many-to-one relationship where each user can have multiple professions, but each user profession entry must reference one user.

1. **User and Address (userr\_address\_fk):**

Each user is linked to one address, with id\_address in the userr table referencing id\_address in the address table. This is a mandatory many-to-one relationship where each address can have multiple users, but each user must reference one address.

1. **User and Manager (Self-Referencing) (userr\_manager\_fk):**

Each user can have one manager, with manager\_id in the userr table referencing id\_user in the same table. This is an optional many-to-one hierarchical relationship where each manager can have multiple users under them.

1. **Work Record and Task (work\_record\_task\_fk):**

Each work record can be associated with one task, with id\_task in the work\_record table referencing id\_task in the task table. This is an optional many-to-one relationship where each task can have multiple work records, but each work record can reference at most one task.

1. **Work Record and User (work\_record\_userr\_fk):**

Each work record is associated with one user, with id\_user in the work\_record table referencing id\_user in the userr table. This is a mandatory many-to-one relationship where each user can have multiple work records, but each work record must reference one user.

1. **Work Record History and Work Record (work\_history\_work\_record\_fk):**

Each work record history entry links to one work record, with id\_work\_record referencing id\_work\_record in the work\_record table. This is a mandatory many-to-one relationship where each work record can have multiple history entries, but each history entry must reference one work record.

1. **Work Record History and User (work\_record\_history\_userr\_fk):**

Each work record history entry is associated with one user, with id\_user in the work\_record\_history table referencing id\_user in the userr table. This is a mandatory many-to-one relationship where each user can have multiple work record history entries, but each entry must reference one user.

# 

# **DD S03 L02**

**Nakreslete ER diagram dle konvencí**

Obrázok, na ktorom je text, snímka obrazovky, diagram, štvorec

Automaticky generovaný popis

# 

# **DD S30 L04**

Maticový diagram se vztahy, nakreslete pro vaše řešení

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | address | certification | client | client\_task | feedback | materials | materials\_used | person | place\_of\_work | profession | task | user\_profession | userr | work\_record | work\_record\_history |
| address | PK | - | - | - | - | - | - | - | 1:M | - | - | - | - | - | - |
| certification | - | PK | - | - | - | - | - | - | - | 1:M | - | - | - | - | - |
| client | - | - | PK | 1:M | - | - | - | 1:1 | - | - | 1:M | - | - | - | - |
| client\_task | - | - | M:1 | PK | - | - | - | - | - | - | M:1 | - | - | - | - |
| feedback | - | - | M:1 | - | PK | - | - | - | - | - | - | - | M:1 | - | - |
| materials | - | - | - | - | - | PK | 1:M | - | - | - | - | - | - | - | - |
| materials\_used | - | - | - | - | - | M:1 | PK | - | - | - | - | - | - | M:1 | - |
| person | - | - | 1:1 | - | - | - | - | PK | - | - | - | - | - | . | - |
| place\_of\_work | 1:M | - | - | - | - | - | - | - | PK | - | - | - | - | 1:M | - |
| profession | - | 1:M | - | - | - | - | - | - | - | PK | - | - | - | - | - |
| task | - | - | M:1 | M:1 | - | - | - | - | - | - | PK | - | - | - | - |
| user\_profession | - | - | - | - | - | - | - | - | - | M:1 | - | PK | M:1 | - | - |
| userr | 1:M | - | - | - | M:1 | - | - | 1:1 | - | M:1 | M:1 | M:1 | PK | - | - |
| work\_record | - | - | - | - | - | - | M:1 | - | 1:M | - | M:1 | - | M:1 | PK | - |
| work\_record\_history | - | - | - | - | - | - | - | - | - | - | - | - | M:1 | M:1 | PK |

1. **DD S04 L01**

Supertypy a podtypy – definujte minimálně jeden případ supertypu a subtypu ve vašem projektu

Obrázok, na ktorom je text, snímka obrazovky, diagram, rad

Automaticky generovaný popisObrázok, na ktorom je text, snímka obrazovky, rad, písmo

Automaticky generovaný popis

# **DD S04 L02**

Popis obchodních pravidel k vašemu projektu

**Userr Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| id\_user | INTEGER | Primary Key | No | Automatically incremented PK |
| skills | VARCHAR(500) |  | Yes | User’s skills |
| password\_user | VARCHAR(30) |  | No | User’s password |
| role\_user | CHAR(1) |  | No | User’s role |
| id\_profession | INTEGER | Foreign Key | No | Profession |
| id\_address | INTEGER | Foreign Key | No | Address |
| manager\_id | INTEGER | Foreign Key | Yes | Manager |

**Address Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_address | INTEGER | Primary | NO | Automatically incremented PK |
| Address\_name | VARCHAR(50) |  | NO | Name of the address |
| Address\_number | INTEGER |  | NO | Number of the address |
| Address\_city | VARCHAR2(40) |  | NO | City of the address |
| Address\_country | VARCHAR2(40) |  | NO | Country of the address |
| Postal\_Code | CHAR(5) |  | NO | Postal code of the address |

**Certification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| id\_certification | INTEGER | Primary | NO | Automatically incremented PK |
| id\_profession | INTEGER | Foreign | NO | Profession |
| Certification\_name | VARCHAR(100) |  | NO | Certification |

**Client Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_client | INTEGER | Primary | NO | Automatically incremented PK |
| Company\_name | VARCHAR2(30) |  | YES | Name of the company |

**Client\_Task Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_Client | INTEGER | Primary | NO | Client |
| Id\_task | INTEGER | Primary | NO | Task |
| role | VARCHAR(50) |  | NO | Role of the client |

**Feedback Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_feedback | INTEGER | Primary | No | Automatically incremented PK |
| Feedback\_text | VARCHAR(500) |  | No | Text of the feedback |
| Id\_client | INTEGER | Foreign | Yes | Client |
| Id\_user | INTEGER | Foreign | Yes | User |

**Materials Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_materials | INTEGER | Primary | NO | Automatically incremented PK |
| Material\_name | VARCHAR(50) |  | NO | Name of the material |
| Material\_description | VARCHAR(255) |  | NO | Description of the material |
| Material\_quantity | INTEGER |  | NO | Quantity of the material |

**Materials\_Used Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| quantity\_used | INTEGER |  | NO | Quantity of material |
| Id\_work\_record | INTEGER | Primary/Foreign | NO | Work record |
| Id\_materials | INTEGER | Primary/Foreign | NO | Materials |

**Person Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_person | INTEGER | Primary | NO | Automatically incremented PK |
| First\_name | VARCHAR(30) |  | YES | First name of the person |
| Last\_name | VARCHAR(40) |  | YES | Last name of the person |
| Email | VARCHAR(40) |  | NO | Email of the person |
| Phone\_number | VARCHAR(13) |  | NO | Phone number of the person |

**Place\_Of\_Work Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_place\_of\_work | INTEGER | Primary | NO | Automatically incremented PK |
| Construction\_name | VARCHAR(60) |  | NO | Name of the construction site |
| Id\_adress | INTEGER | Foreign | NO | Address |

**Profession Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_profession | INTEGER | Primary | NO | Automatically incremented PK |
| Profession\_name | VARCHAR(40) |  | NO | Name of the profession |

**Task Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_task | INTEGER | Primary | NO | Automatically incremented PK |
| Task\_name | VARCHAR(40) |  | NO | Name of the task |
| Task\_description | VARCHAR(200) |  | YES | Description of the task |
| Acutal\_status | VARCHAR(100) |  | YES | Current status of the task |
| Date\_of creation | DATE |  | NO | Creation date of the task |
| Date\_of\_completion | DATE |  | NO | Completion date of the task |
| Comments | VARCHAR(100) |  | YES | Comments about the task |
| Id\_user | INTEGER | Foreign | NO | User |
| Id\_client | INTEGER | Foreign | NO | Client |
| Parent\_task\_id | INTEGER | Foreign | YES | Task |

**User\_profession Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_user | INTEGER | Primary/Foreign | NO | User |
| Id\_profession | INTEGER | Primary/Foreign | NO | Profession |

**Work\_record Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Id\_work\_record | INTEGER | Primary | NO | Automatically incremented PK |
| Description | VARCHAR(500) |  | YES | Description of the work record |
| Date\_work\_Record | DATE |  | YES | Date of the work record |
| Id\_user | INTEGER | Foreign | NO | User |
| Id\_task | INTEGER | Foreign | YES | Task |
| Id\_place\_of\_work | INTEGER | Foreign | NO | Place of work |

**Work\_record\_history Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **Key type** | **Null** | **Description** |
| Modified\_at | DATE |  | NO | Date of modification |
| Old\_description | VARCHAR(2000) |  | NO | Old Description |
| New\_description | VARCHAR(2000) |  | NO | New Description |
| Id\_work\_record | INTEGER | Primary/Foreign | NO | Work record |
| id\_user | INTEGER | Foreign | NO | User |

# **DD S05 L01**

In the project, there are those relationships:

|  |  |
| --- | --- |
| **Task to Parent Task (Self-Referencing)** | **Transferable** |
| **Userr to Address** | **Non-Transferable** |
| **Userr to Profession** | **Non-Transferable** |
| **Userr to Person** | **Non-Transferable** |
| **Userr to Manager** | **Transferable** |
| **Certification to Profession** | **Non-Transferable** |
| **Client to Person** | **Non-Transferable** |
| **Client\_Task to Client** | **Non-Transferable** |
| **Client\_Task to Task** | **Transferable** |
| **Feedback to Client** | **Transferable** |
| **Feedback to Userr** | **Transferable** |
| **Materials\_Used to Materials** | **Non-Transferable** |
| **Materials\_Used to Work\_Record** | **Transferable** |
| **Place\_Of\_Work to Address** | **Non-Transferable** |
| **Task to Client** | **Non-Transferable** |
| **Task to Userr** | **Non-Transferable** |
| **User\_profession to Profession** | **Non-Transferable** |
| **Work\_Record to Userr** | **Non-Transferable** |
| **Work\_Record to Task** | **Non-Transferable** |
| **Work\_Record to Place\_Of\_Work** | **Non-Transferable** |
| **Work\_Record\_History to Work\_Record** | **Non-Transferable** |
| **Work\_Record\_History to Userr** | **Non-Transferable** |

# **DD S05 L03**

Mějte ve svém projektu minimálně jednu vazbu M: N bez informace a jednu vazbu M:N s informací

In the project, you can find these types of M: N relations:

|  |  |
| --- | --- |
| **Name of relation** | **Type of relation** |
| **Client\_task** | **M:N with additional information** |
| **User\_profession** | **M:N without additional information** |

# **DD S06 L01**

*Zakomponujte do svého projektu minimálně jednu identifikující vazbu 1: N s tím, že přenesený cizí klíč bude i klíčem v tabulce nové*

I have used these 1: N relations identifying.

|  |  |
| --- | --- |
| **Name of relation** | **Type of relation** |
| **Certification\_profession\_id\_pk** | **1: N identifying** |
| **Materials\_used\_pk** | **1: N identifying** |

# **DD S06 L02-04**

* *Mějte své schéma v první normální formě – bez neatomických atributů*
* *Ve druhé normální formě – bez vazeb na podklíči*
* *Ve třetí normální formě – bez vazeb mezi sekundárními atributy*

Schema Evaluation:

Let's verify this by listing each table and ensuring compliance with 1NF, 2NF, and 3NF:

**1. Address Table**

1NF: Atomic values.

2NF: No partial dependency since id\_address is the primary key.

3NF: No transitive dependency.

**2. Certification Table**

1NF: Atomic values.

2NF: Composite key (id\_profession, id\_certification) ensures no partial dependency.

3NF: No transitive dependency.

**3. Client Table**

1NF: Atomic values.

2NF: No partial dependency since id\_client is the primary key.

3NF: No transitive dependency.

**4. Client Task Table**

1NF: Atomic values.

2NF: Composite key (id\_client, id\_task) ensures no partial dependency.

3NF: No transitive dependency.

**5. Feedback Table**

1NF: Atomic values.

2NF: No partial dependency since id\_feedback is the primary key.

3NF: No transitive dependency.

**6. Materials Table**

1NF: Atomic values.

2NF: No partial dependency since id\_materials is the primary key.

3NF: No transitive dependency.

**7. Materials Used Table**

1NF: Atomic values.

2NF: Composite key (id\_work\_record, id\_materials) ensures no partial dependency.

3NF: No transitive dependency.

**8. Person Table**

1NF: Atomic values.

2NF: No partial dependency since id\_person is the primary key.

3NF: No transitive dependency.

**9. Place of Work Table**

1NF: Atomic values.

2NF: No partial dependency since id\_place\_of\_work is the primary key.

3NF: No transitive dependency.

**10. Profession Table**

1NF: Atomic values.

2NF: No partial dependency since id\_profession is the primary key.

3NF: No transitive dependency.

**11. Task Table**

1NF: Atomic values.

2NF: No partial dependency since id\_task is the primary key.

3NF: No transitive dependency.

**12. User Profession Table**

1NF: Atomic values.

2NF: Composite key (id\_user, id\_profession) ensures no partial dependency.

3NF: No transitive dependency.

**13. Userr Table**

1NF: Atomic values.

2NF: No partial dependency since id\_user is the primary key.

3NF: No transitive dependency.

**14. Work Record Table**

1NF: Atomic values.

2NF: No partial dependency since id\_work\_record is the primary key.

3NF: No transitive dependency.

**15. Work Record History Table**

1NF: Atomic values.

2NF: Composite key (id\_work\_record, id\_user) ensures no partial dependency.

3NF: No transitive dependency.

Given this evaluation, my database schema satisfies the conditions for 1NF, 2NF, and 3NF.

# **DD S07 L01**

*Vyzkoušejte ve vašem projektu definovat ARC (lze definovat v ORACLE SQL Developeru Data Modeleru)*

Obrázok, na ktorom je text, snímka obrazovky, diagram, rad

Automaticky generovaný popis

# **DD S07 L02**

*Zkuste ve vašem projektu definovat hierarchické a rekurzivní relace*

* Hierarchická relace

Obrázok, na ktorom je text, snímka obrazovky, diagram, rad

Automaticky generovaný popis

The userr table has a hierarchical structure because it contains a manager\_id column that refers back to the id\_user column within the same table. This establishes a parent-child relationship where each user can have a manager, creating a hierarchy.

* Rekurzívna relace

Obrázok, na ktorom je text, snímka obrazovky, diagram, rad

Automaticky generovaný popis

# **DD S07 L03**

**Popište, jak ve vašem systému evidujete historická data**

In my project, I monitor the changes in these entities with regard to the time the changes were made.

* Work\_record\_history
* Work\_record

1. **DD S09 L01**

**Demonstrujte na vašem projektu ukládání změn v čase**

As an example from my project, I can present the entity work\_record\_history in which old descriptions are recorded.

Obrázok, na ktorom je text, snímka obrazovky, písmo, číslo

Automaticky generovaný popis

# **DD S09 L02**

Journaling or logging is implemented through the work\_record\_history table. This table records changes to the work\_record table, effectively keeping a journal of modifications. The table includes timestamps of changes, the user who made the changes, and the old and new descriptions of work records. This allows for tracking and auditing changes over time, which is a form of journaling.

1. Timestamp (modified\_at): Records the exact date and time when the change was made.

2. User Tracking (id\_user): Identifies which user made the change.

3. Change Tracking (old\_description and new\_description): Stores the previous and updated descriptions of the work record.

When a change is made to a work record, an entry is created in the work\_record\_history table. This entry logs the modification, ensuring that any updates are traceable and can be audited.

1. **DD S10 L01**

Revidujte váš návrh podle konvencí pro čitelnost vašeho schématu  
  
Userr Table

Obrázok, na ktorom je text, rad, diagram, rovnobežný

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, písmo, rovnobežný

Automaticky generovaný popis**Sub-Diagram – Relationships among entity wokr\_record and materials and materials\_used**

Obrázok, na ktorom je text, diagram, snímka obrazovky, rovnobežný

Automaticky generovaný popisSubdiagram- realtionships among task, client, person and client\_task

Obrázok, na ktorom je text, snímka obrazovky, rad, písmo

Automaticky generovaný popisSub-diagram – relationships between place of work and address

Obrázok, na ktorom je text, diagram, snímka obrazovky, rad

Automaticky generovaný popisSub-diagram – relationships between user, work\_Record and work\_record history

Obrázok, na ktorom je text, snímka obrazovky, diagram, písmo

Automaticky generovaný popisSub-diagram – relationships between userr, client, feedback

# **DD S10 L02**

Generické modelování – zvažte, případně popište nebo použijte ve svém řešení generický model datových struktur, v čem je tento přístup výhodnější oproti tradičním metodám návrhu datových struktur

Generic modeling provides a flexible and scalable approach to database design. By incorporating generic entities and polymorphic associations, your schema can handle a wider variety of data types and relationships, making it more adaptable to future changes. This approach simplifies the schema, reduces redundancy, and enhances reusability, which are critical for maintaining and scaling complex databases.

# **DD S11 L01**

Popište na vašem projektu příklady integritních omezení pro entity, vazby, atributy a uživatelsky definované integrity

**Entity Integrity**

* Primary Key Constraints: Ensure that each entity (record) can be uniquely identified.
* Example: id\_user in userr table, id\_profession in profession table.

**Referential Integrity**

* Foreign Key Constraints: Ensure that a foreign key value always points to an existing, valid record in another table.
* Example: id\_profession in userr references profession.

**Attribute Integrity Not Null Constraints:**

* Ensure that certain attributes cannot be null. Example: email in userr.

**User-Defined Integrity**

* Unique Constraints: Ensure that an attribute has unique values.
* Example: email in userr.
* Check Constraints: Custom rules to enforce specific business rules.
* Example: Ensuring role\_user in userr is either 'Z' or 'V'

# 

# **DD S11 L02-04**

Obrázok, na ktorom je text, diagram, plán, nalepovacie lístočky

Automaticky generovaný popis*Generujte relační schéma z vašeho konceptuálního modelu a uvědomte si změny, které ve schématu nastaly a proč*