Design, normal forms, versioning

Seminar 5

Conceptual design

1. We define the subject area with which we will work. For example, the subject area of retail.
2. We break it down into non-detailed entities. For example, a purchase receipt, a product, a store, an assortment matrix, etc.
3. We determine how entities will be related to each other: one to one, one to many, etc.
4. We build a visual picture in the ER notation "Crow's Foot" (Crow's Foot) without explicit attributes.

Notation "Crow's foot"

- The entity is represented as a rectangle containing its name.
- The attributes of the entity are written inside the rectangle representing the entity.
- The connection is represented by a line that connects the two entities involved in the relationship.
- The multiplicity of the connection is depicted in the form of a fork. The optional connection is marked with a circle.

Logical design

1. Split non-detailed entities into detailed ones according to the selected data model and normalization.
2. Define attributes.
3. Define relationships between entities according to attributes.
4. Build a visual picture in the ER notation "Crow's Foot" (Crow's Foot) with explicit attributes.

Physical design

Physical design is the creation of a database schema for a specific DBMS.

Physical design of a particular DBMS may include restrictions on the naming of database objects, restrictions on supported data types, etc.

Physical design of a particular DBMS includes the choice of solutions related to the physical storage environment (the choice of disk memory management methods,

the division of the database by files and devices, data access methods), creating indexes, etc ..

The result of the physical design of a logical scheme can be an SQL script and/or a table with a description for each entity.

Example of physical design

```
CREATE TABLE IF NOT EXISTS STUDENT (
    STUDENT_ID   INTEGER      PRIMARY KEY,
    STUDENT_NAME VARCHAR(128) NOT NULL,
    ...
);
...
```

Name	Description	Data type	Restriction
STUDENT ID	Identificator	INTEGER	PRIMARY KEY
STUDENT NAME	Student name	VARCHAR(128)	NOT NULL
...	...	...	...

ER charts

The Entity-Relationship Model (ER-model) is one of the most well-known and widely used models of semantic modeling, developed by P. Chen in 1976.

The main elements of the ER model:

- entity is an item that can be identified in some way that distinguishes it from other items.
- Attribute is a property of an entity (as a rule, atomic).
- key attribute is a unique attribute that uniquely identifies an instance of an entity.
- relationship is an association established between entities. The degree of connection is the number of related entities.

ER charts

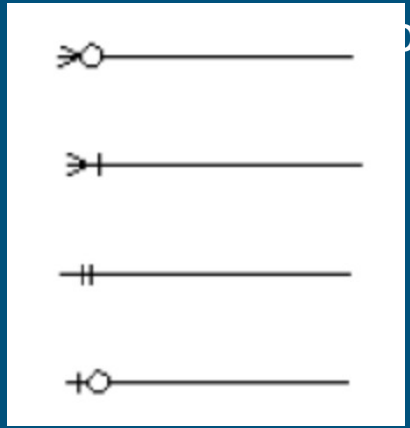
The following notations for ER diagrams exist:

- classical P. Chen notation;
- IDEF notation (Integration Definition for Information Modeling);
- Ch. notation. Bachman;
- notation by J. Martin ("crow's feet");
- notation by J.-R. Abrial (minmax);
- UML class diagrams.

ER charts

When designing an ER model, it is important to distinguish the types of binary connections. There are four types of connections:

- zero or more;
- one or more;
- only one;
- zero or one.



Normal forms

Normal form:

- the list of requirements that we impose on the relation so that it is in a certain SF.
- a property of a relation in a relational data model that characterizes it in terms of redundancy, potentially leading to logically erroneous results of sampling or data modification.

Database normalization — bringing the database to its normal form;

Normal forms

Designed for:

- Minimizing logical redundancy
- Reducing potential inconsistency, eliminating update anomalies
- Development of a database that is:
  - a qualitative representation of the real world
  - intuitive
  - easily scalable
- Simplifying the application of integrity constraints

Normal forms

Not intended for:

- DB performance Changes;
- Changes in the physical volume of the database.

It is produced by decomposition (decomposition of the original variable of the relation into several equivalent) relations:

- lossless decomposition (correct) — reversible decomposition.

Benefits:

- Allows beginners to design good databases.
- Helps beginners to avoid typical mistakes when designing a database.
- Forms the habit of not leaving underlying errors in design to fix later.
- Gradually develops the skill of designing, without relying too much on normal forms.

Anomaly

Anomaly — a situation in the database table that has arisen due to excessive duplication in the table, such that:

- The functioning of a database is significantly affected;
- There are inconsistencies in the database.

Normal forms

First normal form (1NF):

- The values of all attributes of the relationship are atomic (all attributes are simple, contain only scalar values):

Группа	Студент
291	Шехтер
293	Гусев, Медведева, Меркурьева, Шапошников
298	Мавлютов

↓

Группа	Студент
291	Шехтер
293	Гусев
293	Медведева
293	Меркурьева
293	Шапошников
298	Мавлютов

Second normal form (2NF):

- 1NF (atomic relation of attributes) + each non-key attribute is minimally functionally dependent on a potential key (1NF can always be reduced to 2NF, and this

Here, the author of the words and the composer depend only on the fields “Band name” and “Song name”, that is, on a subset of PK (the fact that the song is included in another CD, the author of the words and the composer will not change)

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

↓

Название группы	Название CD-диска	Название песни
Scorpions	World Wide Live	Countdown
Scorpions	World Wide Live	Coming Home
Scorpions	World Wide Live	Blackout
Scorpions	Blackout	Blackout
The Big City	Blackout	Blackout

Название группы	Название песни	Автор слов	Композитор
Scorpions	Countdown	Klaus Meine	Matthias Jabs
Scorpions	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Rudolf Schenker	Klaus Meine

Third normal form (2NF):

- 2NF (atomic relation of attributes, minimal functional dependence on PK) + each non-key attribute is non-transitively functionally dependent on the primary key (each non-key field must contain information about the key, the full

Functional dependencies in this database:

- Employee → Department
- Department → Phone
- Employee →Phone

Сотрудник	Отдел	Телефон
Иванов	Бухгалтерия	11-22-334
Петров	Бухгалтерия	11-22-334
Сидоров	Снабжение	22-33-445

↓

Отдел	Телефон
Бухгалтерия	11-22-334
Снабжение	22-33-445

Сотрудник	Отдел
Иванов	Бухгалтерия
Петров	Бухгалтерия
Сидоров	Снабжение

Versioning

SCD (Slowly Changing Dimension) — rarely changing dimensions, that is, dimensions whose non-key attributes tend to change over time.

There are five main types:

- SCD 0: After entering the table, the data never changes — it does not support versioning. Almost never used.
- SCD 1: Data is overwritten (always up—to-date) - used if the history is not needed.
  - Dignities:
    - Does not add redundancy;
    - Very simple structure.
- Disadvantages:
  - Does not store history.

Versioning

SCD 2: we store the history using two fields (start date of the action, end date of the action: valid_from, valid_to) — used most often.

- Dignities:
  - Stores a complete and unlimited version history;
  - Convenient and easy access to the data of the required period.
- Disadvantages:
  - Provokes redundancy or the establishment of additional tables for storing mutable attributes.

Instead of NULL, some constant is used (for valid_to — '9999-12-31'), which allows you to write a simple condition: WHERE day_dt BETWEEN valid_from_dttm AND valid_to_dttm

Instead of

```
WHERE day_dt >= valid_from_dttm
    AND (day_dt <= valid_to_dttm
        OR valid_to_dttm IS NULL)
```

Versioning

SCD 3: We store the old value for the selected attributes in separate fields. When new data is received, the old data is overwritten with the current values.

- Dignities:
  - Small amount of data;
  - Easy and quick access to the history.
- Disadvantages:
  - Limited history.

Versioning

SCD 4: We store only up-to-date data and the history of changes in a separate table in the main table. The main table is always overwritten with the current data and the old data is moved to another table. Usually this type is used to audit changes or create archive tables.

- Dignities:
  - Fast work with current versions
- Disadvantages:
  - Splitting a single entity into different tables

Task

Design conceptual databases for the following subject areas below. Pre-select the normal form, think about versioning.

- MIPT. Key entities: phys-tech school, department, group, student, teacher, position, audience,class schedule.
- Retail chain stores. Key entities: store, employee, product, receipt, loyalty card, cash register.

Describe the designed models using the SQL query language.