Relational Databases

Database Design

Topics

- Data Normalisation
  - *What are benefits of normalisation?*
  - *Which normalisation form is preferred?*
- Typical Relations
  - One-to-One
    - *Why do we need it?*
    - *What are potential issues with creation?*
  - One-to-Many
    - *How Foreign Keys should be built?*
    - *Foreign Key and NULL values*
  - Many-to-Many
    - *How it is implemented by default?*
    - *Pros and Cons of arrays instead of 3rd table?*
    - *How primary key in 3rd table is built?*
- Denormalisation
  - *When denormalization is done?*
  - *Why do we need denormalization?*
  - *What are typical techniques?*

Helpful Links

- Database Normalization Explained in Simple English
- When and How You Should Denormalize a Relational Database

Practical Part

Scenario #1

- Build a tables structure for the following data domain:
  - Licenses (code, expiry_date, limits)
    - Limits consist of limit group, specific limit in this group and value of the limit
    - Limit group is something like “Clients”, “Time”, “Events”
    - Each limit group has its own sub-limits and list of allowed values
      - “Clients” has “Active Clients” and “Accounts” as sublimits
      - “Time” has “xDRs”, “Orig Volume” and “O+T Volume” as sub-limits
        - “xDRs” sublimit has allowed values of 1M, 5M and 10M
        - “Orig Volume” has allowed values 10M min, 20M min
  - Customers (company name)
    - Customers might be referred by other customers
    - Customers might have 1 or more contact persons
  - Contact Persons (full name, email, phone, job title)
- Suggest denormalisation taking into account that:
  - License limits are accessed very often from different places
  - We often need to know how many other customers one has referred

Scenario #2

Business Scope

You are designing database for general initial part of the HRM system. The system will have to operate with the following entities:

- Employee (name, sex, home address, work/private contacts - email, phone)
- Offices (name, address)
- Teams (name)
- Job Titles (name)

The database should include following scenario:

- There is a set of static parameters about employee like sex, name, contact data - this part of data is typically set once and edited not very often
- There is a date-related set of parameters (office, team, job title) which is set on some specific date and then may be edited on specific date, we have to track dates when employee changes office or team and be able to get this parameters for specific date
- There is one more date-related parameter - status which might be “active” when somebody is working and “terminated” if somebody resigned from the company

Here are some examples of date-related usage:

- We may schedule change of the Job Title as of January 1, 2019. But we can do it in advance.
- We may forget about change of somebody’s office, and change it with date in the past.
- We may agree with somebody about resign and schedule it for specific date.

There is one more relation that we have to accomplish - “Reports To”. Each employee reports to somebody which reflects organization structure. It is not always head of the team, as there might be internal relations within the team. This parameter is also date-related.

Task Description

1. Design tables structure for the given business specification
    a. Mark types of the fields
    b. Mark all relations (PKs, FKs, modes)
    c. Mark indexes
2. Mark and describe denormalization applied if any
3. Write query which will show list of current active employees and all their data as of today

Scenario #3

Business Scope

You are designing database for Leaves Management part of the HRM system. The system will have to operate with the following entities:

- Employee (as defined in the previous task)
- Leave Entitlements
  - There might be different types of leaves (medical, paid, unpaid, ...)
  - Leaves may accrue by different periods (daily, monthly, quarterly, annually)
  - Leaves may accrue at the beginning or end of the period
  - For annual accrual the year may be counted as calendar year or year since join date
  - The accrual value (leave amount) is counted in hours
  - There is a limit of leave amount to be carried to the following year
  - Different types of leaves may be counted at different ratio (for example 1 work from home is counted as 4 hours leave)
  - Leaves may be available or not at the probation period
- Employees Leaves
  - Each employee may apply for a leave
  - He may specify when he wants leave and for which duration
  - The application may be approved or rejected
- Leaves Balances
  - We need to be able to see current leave stats (balances) for each employee

The above paragraph explains both - data and potential behavior. However it does not explain exact objects and relations of how this should be implemented. You may ask questions if needed for design decision. You may make assumptions or suggestions for the object model based on given scenarios.

Task Description

1. Design tables structure for the given business specification
   a. Mark types of the fields
   b. Mark all relations (PKs, FKs, modes)
   c. Mark indexes
2. Mark and describe denormalization applied if any