BITAH05 – Database technologies

Jasper Anckaert

Trial exam

Theory

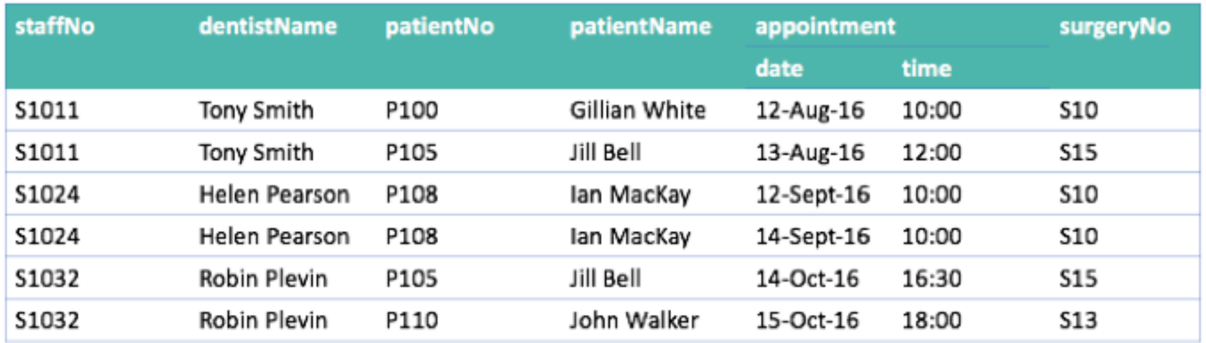
- Questions about:
    - (Relational) Database management systems
    - Column types in MySQL
    - SQL
    - NoSQL systems
    - (Rest) APIs

Exercises

- Exercise 1
  - Create a new folder *exercises* in your home directory
  - Initialize a git repository in this folder
  - Make sure your repository is available on GitHub
  - Add the answers of the following exercises to this repository, each time with a different commit message
  - Save your command in a file called *exercise1.txt* and add this file to your repository

Exercises

- Exercise 2
- A dental practice keeps track of the following data



staffNo	dentistName	patientNo	patientName	appointment		surgeryNo
				date	time	
S1011	Tony Smith	P100	Gillian White	12-Aug-16	10:00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-16	12:00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sept-16	10:00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sept-16	10:00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-16	16:30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-16	18:00	S13

- Normalise this data up to 3NF, taking in to account these functional dependencies:
  - staffNo, apptDate, apptTime → patientNo, patientName
  - staffNo → dentistName
  - patientNo → patientName, surgeryNo
  - staffNo, apptDate → surgeryNo
  - apptDate, apptTime, patientNo → staffNo, dentistName
- Save the results of each normisation step (with description) in a document called exercise2.txt. Add this file to your repository with a clear commit message.

Exercises

- Exercise 3
  - Create an ERD schema with 3 tables
    - Doctor
      - Name
      - Date_of_birth
      - Address
      - phone_number
      - Salary
    - Medical
      - Overtime_rate
    - Specialist
      - Field_area

Exercises

- Exercise 3 – continued
    - Add 2 tables to your schema
        - Patient
            - Name
            - Address
            - Phone_numer
            - Date_of_birth
        - Appointment
            - Date
            - Time
    - Think carefully how you will integrate this with your existing tables

Exercises

- Exercise 3 – continued
  - Add 2 tables to your schema
    - Payment
      - Details
      - Method
    - Bill
      - Total
  - Careful: there is a many-to-many relation between bill and payment
  - Save your .mwb file and add it to your repository
  - Forward enigneer your model
  - Save the SQL code in file called *exercise3.sql* and add this file to your repository.

Exercises

- Exercise 4
  - Create a new database called *vacation*
  - Create tables and insert data based on *vacation.sql*
  - Make sure you have sufficient right on this database
  - Execute following queries and save both your query and the results in a file called *exercise4.txt* and add this file to your repository.

  - List the names and addresses of all guests in London, alphabetically ordered by name.
  - List all double or family rooms with a price below £40.00 per night, cheapest first.
  - List the bookings for which no date_to has been specified.
  - What is the average price of a room?
  - What is the total revenue per night from all double rooms?
  - How many different guests have made bookings for August?

Exercises

- Exercise 4 – continued
  - List the price and type of all rooms at the Grosvenor Hotel.
  - List all guests currently staying at the Grosvenor Hotel.
  - List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is occupied.
  - What is the total income from bookings for the Grosvenor Hotel today?
  - List the rooms that are currently unoccupied at the Grosvenor Hotel.
  - What is the lost income from unoccupied rooms at the Grosvenor Hotel?
  - List the number of rooms in each hotel in London.
  - What is the most commonly booked room type for each hotel in London?
  - Update the price of all rooms by 5%.

Exercises

- Exercise 5

  - Import the restaurant information from *restaurants.json*
  - Execute following queries and save both your query and the results in a file called *exercise5.txt* and add this file to your repository.

  - Display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.
  - Display the second group of 5 restaurants which are in the borough Bronx.
  - Find the restaurants who achieved a score more than 90.
  - Find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.
  - Find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.
  - Find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.
  - Find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

Exercises

- Exercise 5 – continued
  - Find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates.
  - Give a list of all restaurants, ordered by cuisine alphabetically and borough in reverse alphabetical order
  - Write a query to know whether all the addresses contain a street or not.

Exercises

- Exercise 6
  - Using the Ensembl REST API
    - Get the sequence for the region from basepair 32889000 to 32891000 of human chromosome 13 in FASTA format. Hardmask and softmask the sequence. How many repeat regions are there in this sequence?
    - Get the Ensembl Gene ID for the human CCR5 (chemokine (CC motif) receptor 5) gene.
    - Has an orthologue for this gene been identified in chimpanzee?
    - Retrieve the genomic FASTA sequence for *ENST00000001146*
    - Return the archived sequence for both *ENSG00000002587* and *ENSG00000003056*
    - Get a sequence from 250 nucleotides located on human chromosome 7 starting at position 35182
  - Save your curl requests and the results in a file called *exercise8.txt*. Add this file to your repository.