

First Lecture 06-09-2019

Info

- Gurinov, A. Assistant Professor with 20 years of experience in database development
- learn about basic concepts of databases
- one semester now, used to be two, used to have principles and design, now only one
- we now have a more intense version
- first half will be focused on working with data, like queries to databases
- show how data can be sorted and turned into information
- we shall get used to the tools we use
- second half will be on design: how to design tables, separate or combine them etc.
- maybe add some modern stuff if there is enough time
- there will be a syllabus for the course over the weekend
- our model will be auca.kg/e-course for learning
- Relational Databases:
 - stores data in tables
 - it is not a graph, .json,
 - what means organized? Structured
 - every table is a reflection of an object from the real or imaginary world
 - attributes are put into columns
 - we need to define boundaries of attributes, how should the data be stored as data types
 - data about data is called metadata

Environment

- Microsoft SQL Server Management Studio - SSMS
- we can use whatever, but he will not support anything but MS SQL Studio
- Language: SQL

Notes

- What is data?
- anything that can be stored
- information?
- info about and object
- info that can be analysed and accessed
- What is the difference between data and information?
- data is information that is used
- information that is stored and can be accessed
- data is always stored somewhere
- anything that can be analysed is data?
- data is organized information
- Data: its raining today. -> Information: it will rain today at 10pm
- **data is raw, information is organized data, it has more context**
- information is used to solve some problems
- data can be interpreted and used for different kinds of uses
- What is the difference between information and knowledge?
- subset of information?
- knowledge is from the past, but info is from the present?
- knowledge has some use, information maybe not?
 - does not seem like it makes sense?
- knowledge can not be false but information cannot?
- **knowledge is something that is generalized**
- something that is learned
- What is the difference between knowledge and wisdom?

- experience is needed to make wisdom out of knowledge?
- when we know how to use knowledge correctly
- applied knowledge
- make correct use of knowledge
- What is a database?
- collection of data stored somewhere
- is a notebook a database?
- it needs to be organized/sorted?
- **database is a set of organized data that is collects and stores somewhere**
- there are many different definitions of databases
- organized, stored electronically, makes sense in some sense, it can be edited, retrieved, stored, aggregated...
- some stuff about data types, bit, int, datetime, nvarchar (unicode string), numeric(2, 2) (float with two before and after the point), varbinary (just bin data)
- some things are strings, sometimes to guarantee data integrity, one can use something like enums or stuff, so strings are a threat to integrity
- every entry has to be uniquely indentifyable, by a number in the table
- everything just gets an id field which is a number
- primary key: the unique identifier of a table
- foreign key: it is a link between two tables, but it also restricts the possible input to the list of foreign keys, also info whose id is used in other tables can not be deleted from its home table -> integrity
- Data integrity: making sure that the data is not corrupted and is actually usable, be as strict as possible to leave no room for error

Lab

- the instructor is senior database dev, Deryhanskii, G.
- the data types are the same as in general programming
- int is 32 bits
- nvarchar is a variable unicode char, datetime2 is very accurate
- date is just the date, datetime also has time information
- numeric is a float with designated lenghts
- primary key is unique, used to quickly, identify and select entries in the table
- normalization: avoiding redundancy of data, which is not good
- e.g. Peter - Male; Alex - Males; Sasha - Mele; will not work when searching for "Male"
- it should be Peter - 1; Alex - 1; Sasha - 1;
- split tables up from the main one to contain them thematically
- we use different tables to organize things like they should be organized

SSMS

- SQL Server vs. Database:
- SQL manages the bases, the server just hosts the stuff
- SSMS gives a graphical interface that allows the accessing of the server
- we use capitalized camel case for the names
- right-click on database and select "New Database"
- by convention add Db to the end of the name
- now add some tables: new table on the tables tab
- you may enter the column name, data type, allow null for the field
- primary key can be anything, but int is considered to be good practice
- all tables shall have Id as first row and int as data type
- nvarchar is unicode, while varchar is ascii, so it's important to distinguish that
- Department should not be in the student table because it should be normalized
- date of birth instead of age, as age changes, stored as date, as the time is not relevant
- we can separate emails and phone numbers into a separate table to normalize

- whether or not to make a separate AucaId from Id is a question of opinion, although separating them has some advantages
- Ctrl + S to save and name the table
- setting up auto-increment: in identity specification, set identity increment to 1 and seed to 1 or whatever
- Gender table:
 - Id - int, click to make primary key
 - Name - nvarchar
 - click and select “Design” to return to the drafting site
 - then create a row for Gender - int, make it nullable
 - then you may enter null or the selected ids from the Gender table
 - click the gender row, click relationships, add, name it FK_from_to or FK_primary-key-table_foreign-key-table
 - primary key table is the one we take the key from, the foreign key table is the one that holds the foreign keys
- contacts table:
 - uniqueidentifier: xxxx-xxxx-xxxx, which is like int, kinda
 - Name: nvarchar
 - also we need to take StudentId to map the contact info to each student
 - then the data type in the main table must also be uniqueidentifier
 - to show the structure of the database: go to database diagrams, click new database diagram
 - arrow to something: many-to-one relationship
- **HOMEWORK:** create such a database on my own, also see SQL for command line, expand the tables and create new ones. At least 8 tables with different contents
- solveway.club for training in queries etc., start training to actually write queries