Database Systems

Data types,
Creating queries by graphical interface

Bence Molnár

2022.03.22.

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Agenda

- Retrospection
- Data types
- Simple queries
- Joining tables
- Grouping, aggregation
- Limitation of graphical interfaces

Challenges of relational databases

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Bad table

Invalid value

(no candidate key)

Name	Firstname	Age	ID	SSN
John Doe	John	23	1	101
Alba Flores	Alba	-1	2	102
John Doe	John	23	1	101

redundancy

(if there were a key, these fields are functional dependent)

Bad table

Duplicate row

inconsistency

Name	Firstname	Age	ID	SSN
John Doe	John	23	1	101
Alba Flores	Alisa	-1	2	102
John Doe	John	23	1	101

We cannot find a person based on lastname

Schema design

Large to small scale:

- Defining tables considering normalization guidelines
- Defining field attribute types
- Formulating restrictions/rules on field values

Attribute types

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

ATTRIBUTE TYPES

- For each attribute we define a domain (set of possible values), this is the data type
- NULL (empty) value is applied if we do not know the value of a field.
- Common types:
  - *Numerical*
    - Float
    - Integer
    - AutoNumber
  - *Character string*
  - *Logical*
  - *Date*

ATTRIBUTE TYPES

- Compound types (examples):
  - *Mask: 000-000-000, XXXXX*
  - *LineString, sphere, geometric items*
  - *Binary Large Object (BLOB)*
    - Image, MP3, etc...
- Types are included in schema:

Grades(ID: AutoNumber, Name : String, Grade: Integer)

SELECTING PROPER ATTRIBUTE TYPES

- Storage requirements
  - *Like on a sheet of paper: column width is defined by the longest/biggest value in the column*
  - *Multiplicity of values (1-9, a-z, A-Z...)*
- Operations
  - *Math operations*
  - *Sorting*
  - *Equality and difference inspection (redundancy)*
- References
  - *Code tables – fixed set of values*
  - *On an other relation's column item*

Further examples

- Profile(ID: AutoNumber, Name: String, ProfileArea: Real, Moment of inertia: Real, Price: Integer)
- Sphere(ID: AutoNumber, X: Real, Y: Real, Z: Real, R: Real)
- Road(ID, Name, Class)
- RealEstates(ID, ParcelNumber, Owner, Area, Price)
- Define data types of last two relations
- Multiple correct answer may exist

Additional restrictions

- We can define additional restrictions/rules
  - *Only positive number*
  - *Only odd numbers*
  - …
- We can define new data types

Creating queries by graphical interface

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Queries

1. List active students and order based on the specialization!
2. How many attempt did given user (uid=30) made to submit homework (in each of the three homework tasks)?
3. Display all details of the project specified by a given user (uid=30) (messages can be ignored here)!
4. Display activities on the site for every day (based on date)!
5. Display daytime activity (units: hours)
6. List visitors settlements (Hungarians only), order by number of clicks.

Building a query

- Required tables?
- Columns to use/display?
- Is there a need to filter rows?
- Grouping?
- Aggregation?
- Ordering?
- How many records?

List active students and order based on the specialization!

1st query

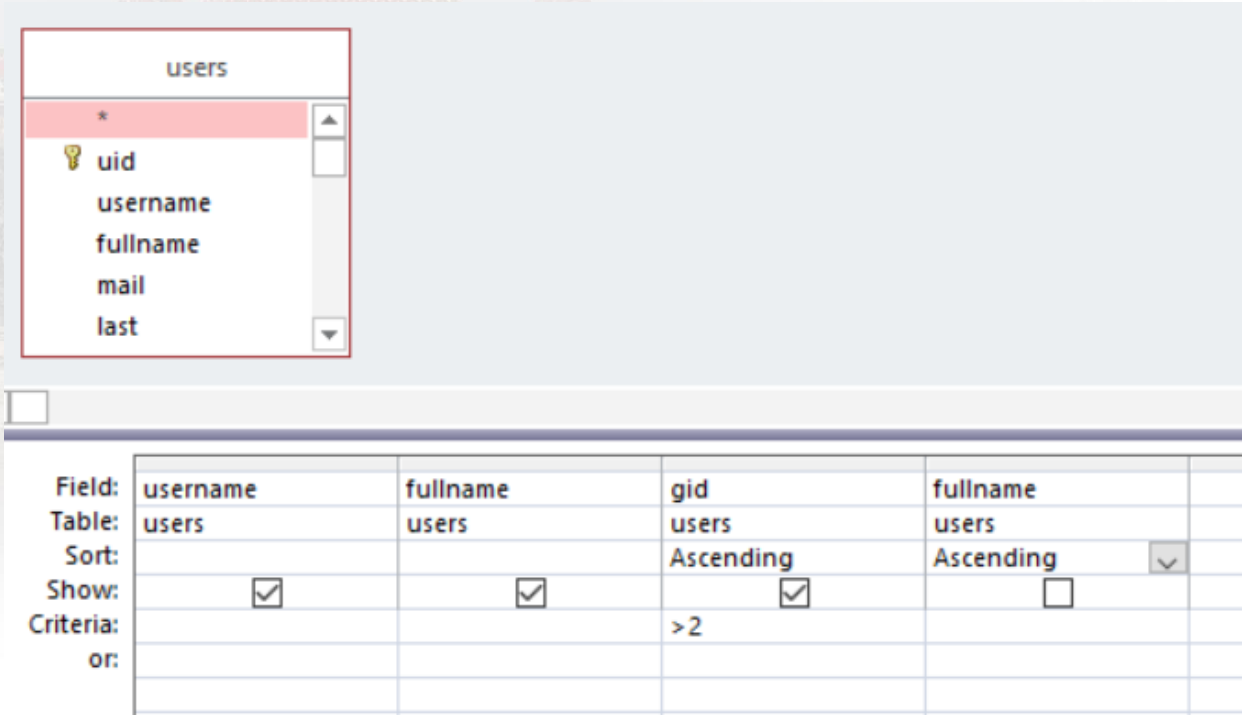
Name	Neptun	Specialization
~~Barna Marianna~~	CF9A74	Inactive
Dr. Antal Olivér	CF9C9F	Inactive
Dr. Szekeres Zoltán	CF9BDF	Inactive
Sipos Gáborné	CF9BE7	Inactive
Varga Olivér	CF9A91	Inactive
Dr. Bakos István PhD	CF9BAD	Infra
ifj. Vass Zéténé	CF99D2	Infra
Balázs Bendegúz PhD	CF9B05	Structural
Bálint Kevin	CF9A4F	Structural
Bálint Kristóf PhD	CF9DF1	Structural

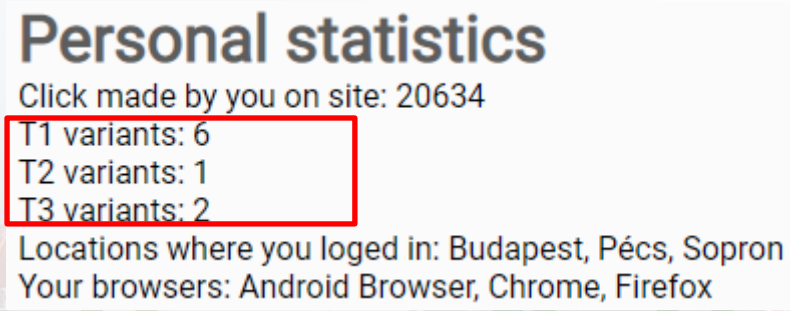
List active students and order based on the specialization!

- Required table: users
- Required columns: username, fullname, gid
- Row filtering: gid>2
- Ordering: gid

- Q: What if we want to order based on specialization names?

List active students and order based on the specialization!





How many attempt did given user (uid=30) made to submit homework (in each of the three homework tasks)?

2nd query

How many attempt did given user (uid=30) made to submit 1st homework?

- Required tables: part1, users
- Required columns: uid, p1id, username
- Grouping: uid, username
- Ordering: Count(p1id)

- Q: How to create this query for all tasks?

How many attempt did given user (uid=30) made to submit in each of the three homework tasks?

- Two possible solutions:
  - *Single row*

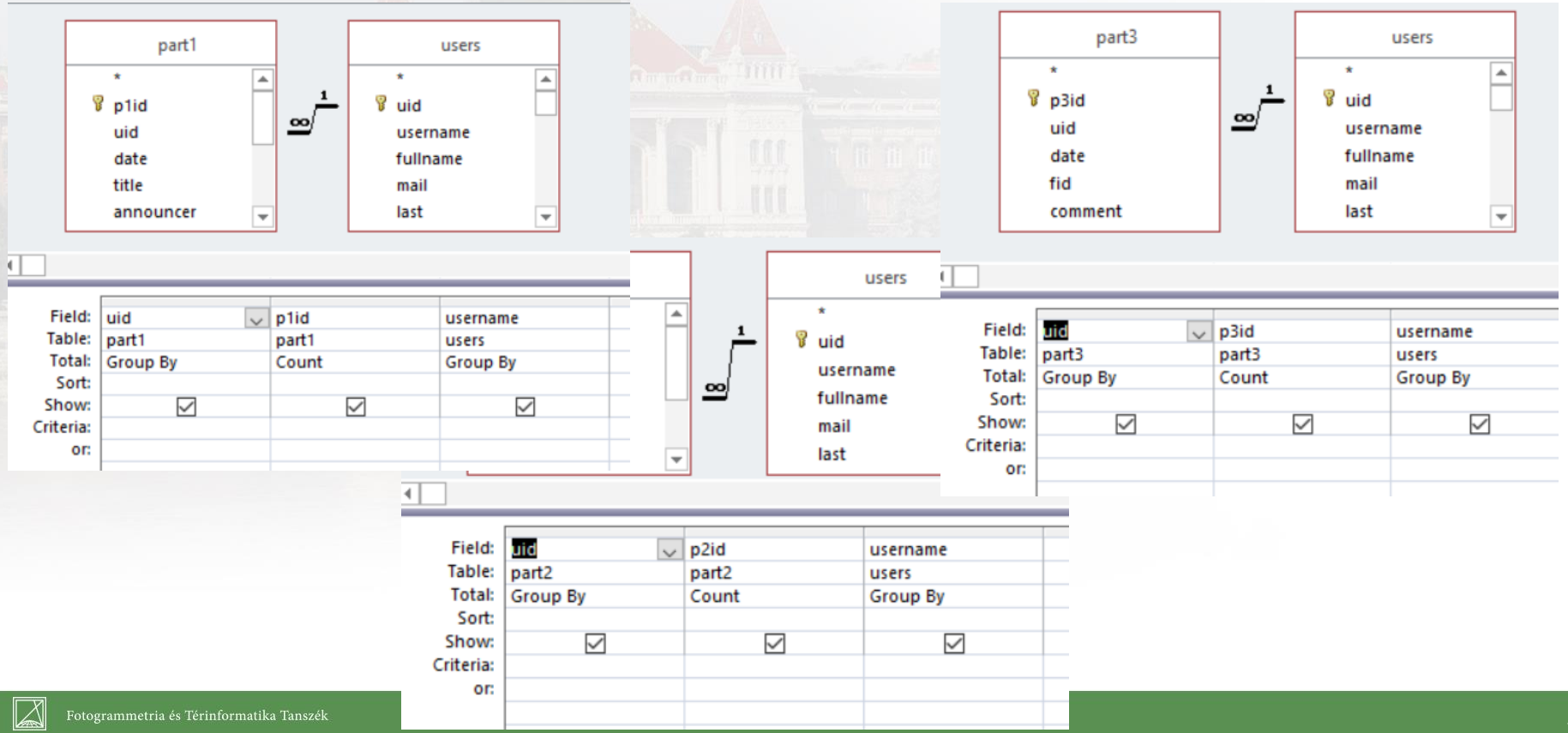
User	1st HWCnt	2nd HwCnt	3rd HWCnt
30	6	1	2

  - *Multiple rows*

User	HW	Count
30	1	6
30	2	1
30	3	2

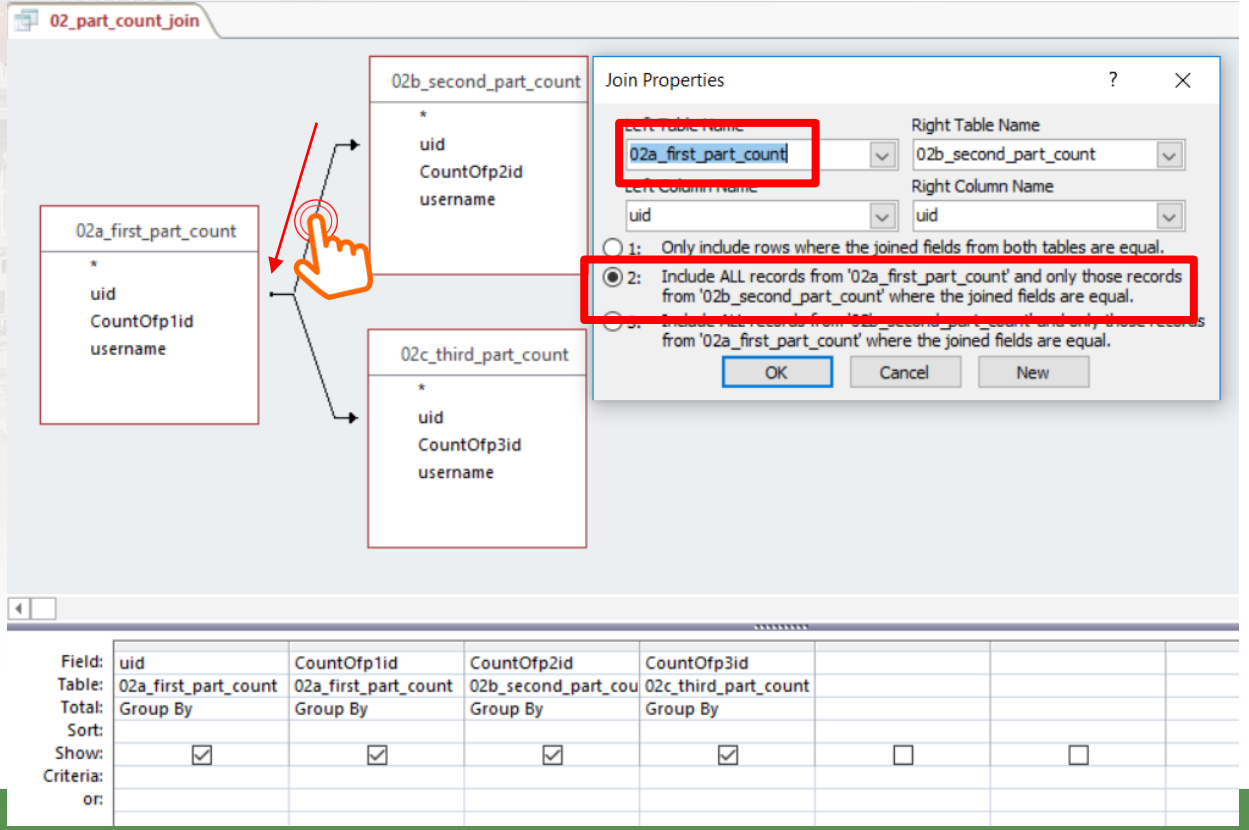
- We will combine multiple queries to achieve the goal

How many attempt did given user (uid=30) made to submit in each of the three homework tasks?



How many attempt did given user (uid=30) made to submit in each of the three homework tasks?

If we want to display information from different tables/queries next to each other; we use JOIN operation.



How many attempt did given user (uid=30) made to submit in each of the three homework tasks?

If we want to display information from different tables/queries under each other; we use UNION operation.

This is not supported on the graphical interface of Access, we use SQL instead.

```
(SELECT 1, uid, CountOfp1id FROM 02a_first_part_count WHERE uid=30) UNION
(SELECT 2, uid, CountOfp2id FROM 02b_second_part_count WHERE uid=30)  UNION
(SELECT 3, uid, CountOfp3id FROM 02c_third_part_count WHERE uid=30);
```

Question of filtering – before or after grouping?

What happens if we want to remove p1id=19 record (this version vas only submitted to send message, no changes on original submission)?

- Column p1id is used for aggregation by applying the COUNT() function. Thus all 19 submissions will be removed and not the submission with p1id=19. (cf9b05 [58])
- Once we add a new column with field p1id and at totals we select WHERE and applying criteria 19, than we achieve expected result. Originally 7 submission will be reduced to 6. (cf99b1 [10])

Display all details of the project specified by a given user (uid=30) (messages can be ignored here)!

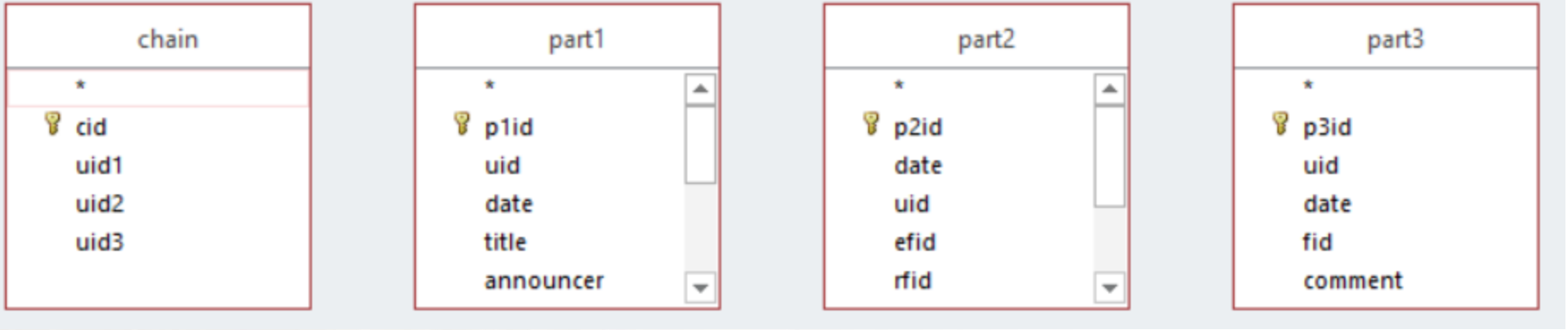
3rd query

Display all details of the project specified by a given user (messages can be ignored here)!

- Required tables: chain, part1, part2, part3
- Required columns: part1.*, part2.*, part3.*
- Row filtering: uid1=30, chain.uid1=part1.uid, chain.uid2=part2.uid, chain.uid3=part3.uid
- Grouping: chain.uid1
- Aggregation: Last() for all displayed columns

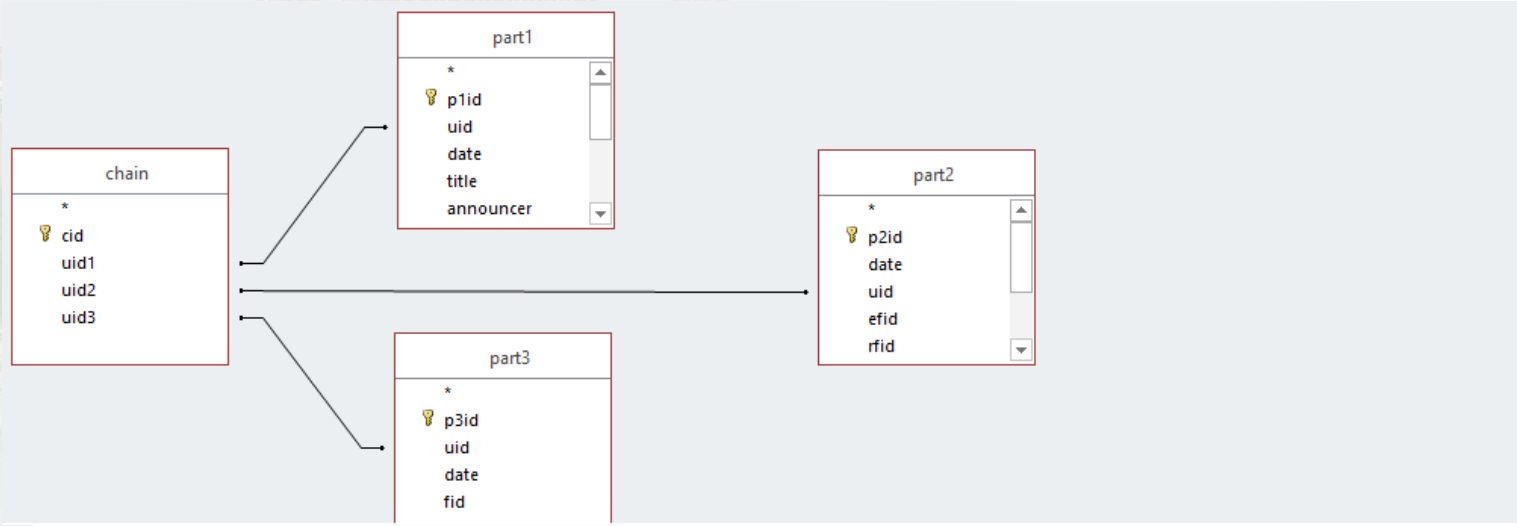
- Q: Do we need users table?

Display all details of the project specified by a given user (messages can be ignored here)!

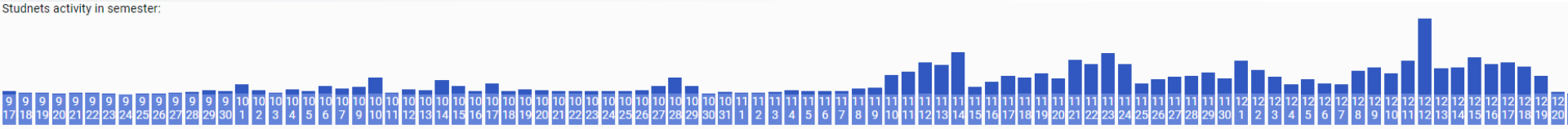


Field:	uid1	uid	uid	uid	date	title	announcer	contractor	task	goal	input	output	comment	fid	date	efid	rfid	xfid	queries	qfid	comment	date	fid	comment
Table:	chain	part1	part2	part3	part1	part1	part1	part1	part1	part1	part1	part1	part1	part1	part2	part2	part2	part2	part2	part2	part2	part3	part3	part3
Total:	Group By	Where	Where	Where	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last
Sort:																								
Show:	☑	☐	☐	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	30	[chain].[uid1]=[part1].[uid]	[chain].[uid2]=[part2].[uid]	[chain].[uid3]=[part3].[uid]																				
or:																								

OTHER SOLUTION



Field:	uid1	date	title	announcer	contractor	task	goal	input	output	comment	fid	date	efid	rfid	xfid	queries	qfid	comment	date	fid	comment
Table:	chain	part1	part1	part1	part1	part1	part1	part1	part1	part1	part1	part2	part2	part2	part2	part2	part2	part2	part3	part3	part3
Total:	Group By	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last	Last
Sort:																					
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	[uid1]=30																				
or:																					



Display activities on the site for every day (based on date)!

4th query

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

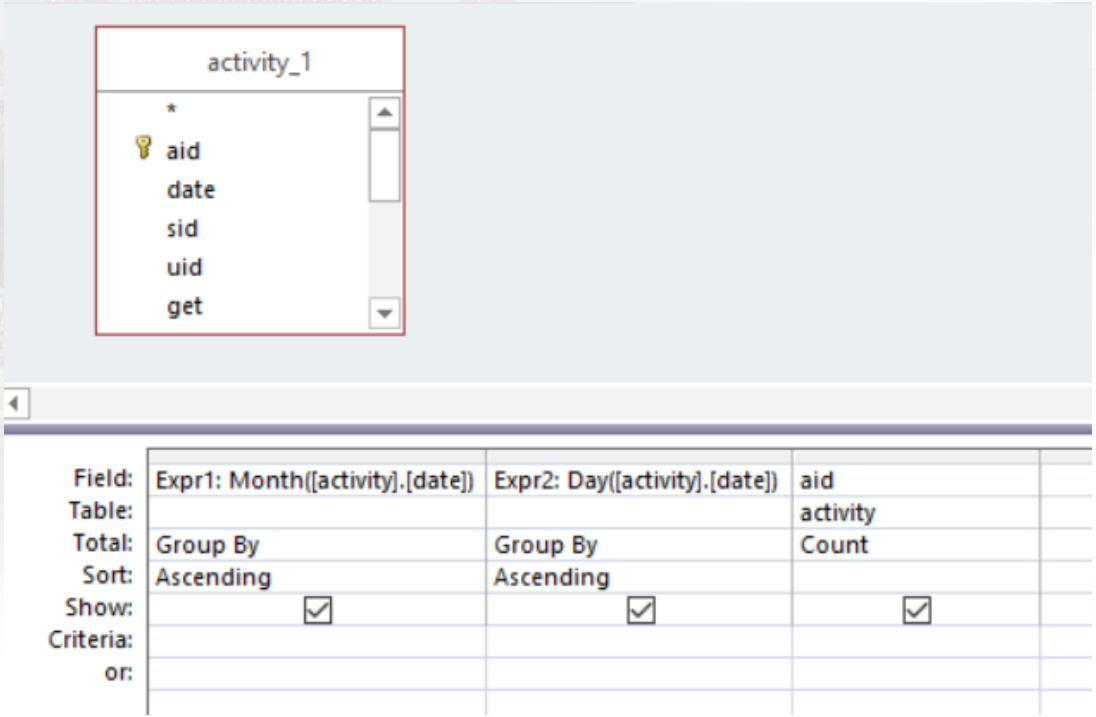
Fotogrammetria és Térinformatika Tanszék

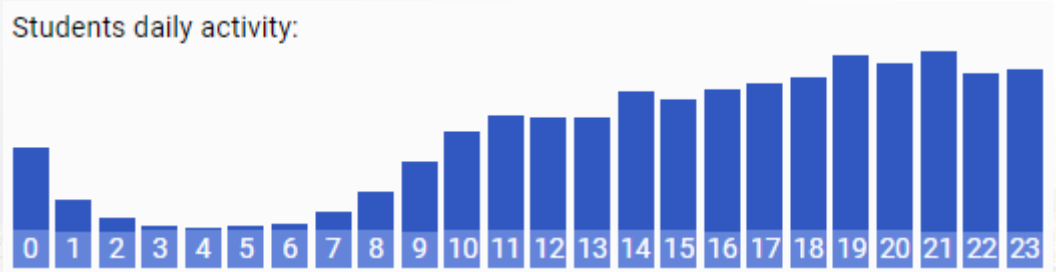
Display activities on the site for every day (based on date)!

- Required tables: activity
- Required columns: Month(date), Day(date), aid
- Grouping: Month(date), Day(date)
- Aggregation: Count(aid)
- Sorting: Month(date), Day(date)

- Q: Is it sufficient to check Day() only? What if different column display and sorting order is required?

Display activities on the site for every day (based on date)!





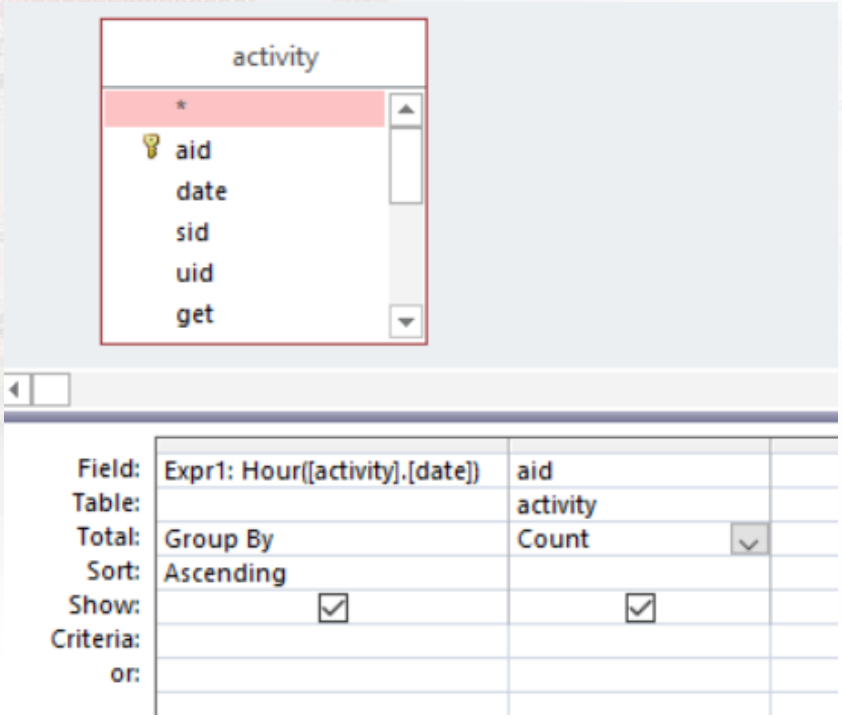
Display daytime activity (units: hours)

5th query

Display daytime activity (units: hours)

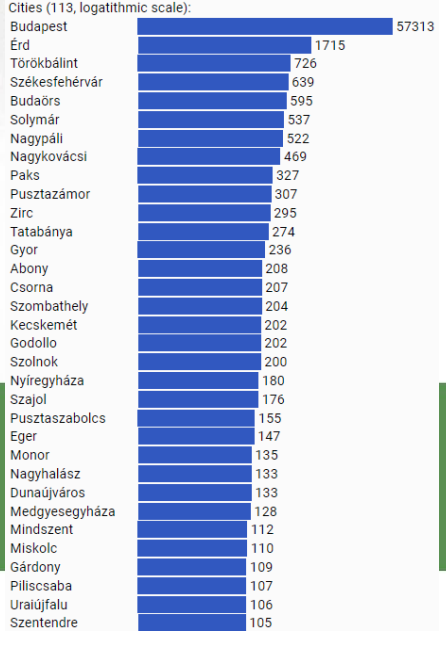
- Required tables: activity
- Required columns: Hour(date), aid
- Grouping: Hour(date)
- Aggregation: Count(aid)
- Ordering: Hour(date)

Display daytime activity (units: hours)



List visitors settlements (Hungarians only), order by number of clicks.

6th query



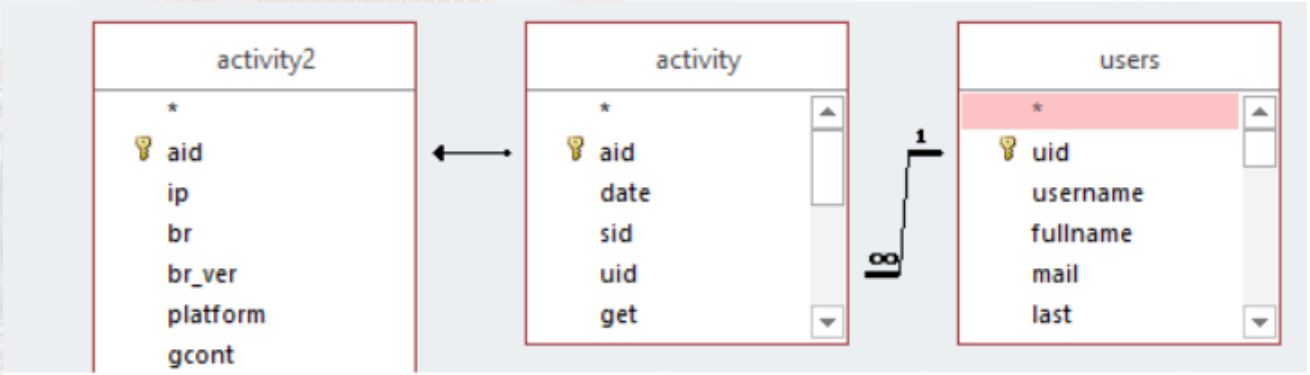
BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

List visitors settlements (Hungarians only), order by number of clicks.

- Required tables: activity, activity2, users
- Required columns: gcity, aid
- Row filtering : gid<>1 AND gcity<>”
- Grouping: gcity
- Aggregation: Count(aid)
- Ordering: Hour(date)

List visitors settlements (Hungarians only), order by number of clicks.



Field:	gcity	aid	gid		
Table:	activity2	activity2	users		
Total:	Group By	Count	Where		
Sort:		Descending			
Show:	☑	☑	☐	☐	
Criteria:	<>''		<>1		
or:					

Conclusion

- Retrospection
- Data types
- Simple queries
- Joining tables
- Grouping, aggregation
- Limitation of graphical interfaces

Thank you for your attention!

Questions?

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék