Database Usage (and Construction)

More SQL Queries and Relational Algebra

Tests on groups

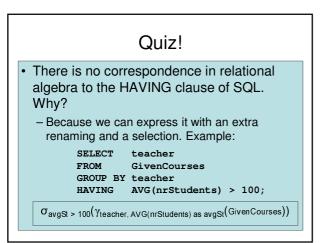
- Aggregations can't be put in the WHERE clause - they're not functions on rows but on groups.
- · Sometimes we want to perform tests on the result of an aggregation.
 - Example: List all teachers who have an average number of students of >100 in their courses.
- · SQL allows us to put such tests in a special HAVING clause after GROUP BY.

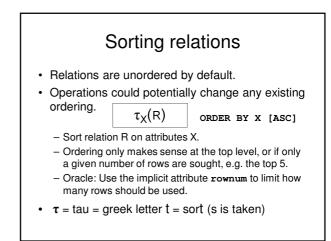
Quiz!

List all teachers who have an average number of students of >100 in their courses.

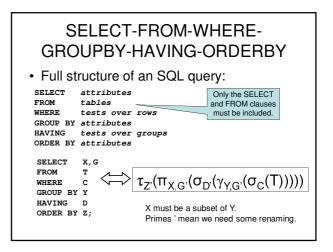
SELECT	teacher		
FROM	GivenCourses		
GROUP BY	teacher		
HAVING	AVG(nrStudents)	>	100;

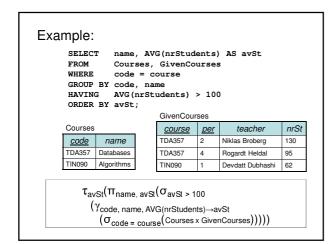
SELECI FROM GROUP	Gi	venCourses		
HAVING	G AV	G(nrStudents)	,	
code	period	teacher	#students	AVG(nrSt.)
	0	Niklas Broberg	130	130
TDA357	2	Ninas Broberg	I	
	2	Devdatt Dubhashi	05	95
TDA357 TIN000 TDA357		<u> </u>	05 135	95





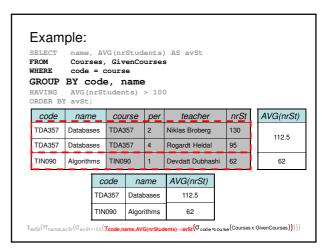
Exa	ample
SELECT FROM ORDER B	* Courses Y name;
<u>code</u>	name
TIN090	Algorithms
TDA357	Databases
TDA590	OOSD

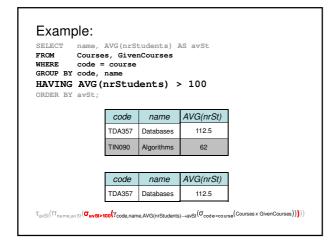




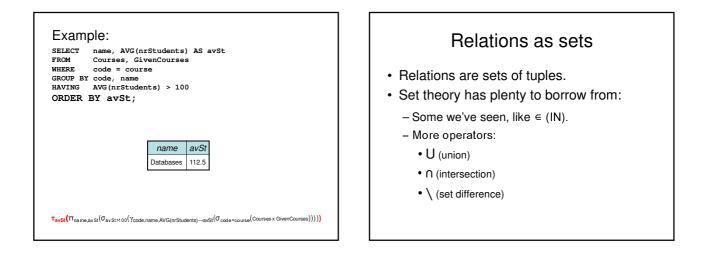
FROM		AVG (nrStu ses, Gi				
AVING	SY code,	= course name rStudents)	> 100			
KDEK E	code	name	course	per	teacher	nrSt
	TDA357	Databases	TDA357	2	Niklas Broberg	130
	TDA357	Databases	TDA357	4	Rogardt Heldal	95
	TDA357	Databases	TIN090	1	Devdatt Dubhashi	62
	TIN090	Algorithms	TDA357	2	Niklas Broberg	130
	TIN090	Algorithms	TDA357	4	Rogardt Heldal	95
	TIN090	Algorithms	TIN090	1	Devdatt Dubhashi	62

SELECT FROM WHERE	Cour	ses, le =	Gi	rStud venCo ours	urse) AS avSt s				
GROUP BY HAVING		/		nts)	> 10	D				
ORDER B	r avSt	<i>'</i>	co	urse	per	teach	er	nrSt	1	
TDA357	Databa		TDA		2	Niklas Brob		130		
TDA357	Databa	ses	TDA	357	4	Rogardt He	Idal	95		
TDA357	Databa	ses	TINC	90	1	Devdatt Dut	ohashi	62		
TIN090	Algorith	60	ode	na	me	course	per	te	acher	nrS
TIN090	Algorith			Datab		TDA357	2	Niklas E		130
TIN090	Algorith	TDA	357	Datab	ases	TDA357	4	Rogard	t Heldal	95
		TIN	090	Algori	thms	TIN090	1	Devdatt	Dubhashi	62



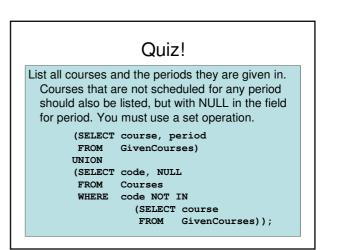


FROM C WHERE C GROUP BY C HAVING A ORDER BY a	ode = cou ode, name VG(nrStud	1		
	code	name	AVG(nrSt)	
	TDA357	Databases	112.5	
		name Databases	<i>avSt</i> 112.5	



Set operations

- · Common set operations in SQL
 - UNION: Given two relations R_1 and R_2 , add them together to form one relation $R_1 U R_2$.
 - INTERSECT: Given two relations R_1 and R_2 , return all rows that appear in both of them, forming $R_1 \cap R_2$.
 - EXCEPT: Given two relations R_1 and R_2 , return all rows that appear in R_1 but not in R_2 , forming $R_1 \setminus R_2$. • Oracle calls this operation MINUS.
- All three operations require that R₁ and R₂ have (almost) the same schema.
 - Attribute names may vary, but number, order and types must be the same.



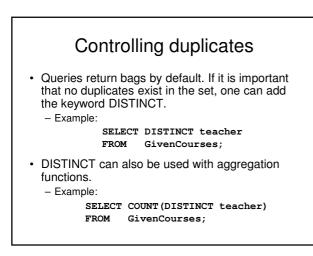
(SELECT coo FROM Giv UNION	de, period venCourses		<u>code</u>	2	name
(SELECT CO	de, NULL		TIN090		Algorithms
	urses		TDA590)	OOS
WHERE CO	de NOT IN CT code		TDA357	,	Databases
FROM		urses));	TDA100	1	AI
•		<pre>ourses));</pre>	TDA100)	AI
•		teacher	TDA100		AI
FROM	GivenCo				udents
FROM	GivenCo period	teacher	berg	#stu	udents
FROM code TDA357	GivenCo period 2	teacher Niklas Bro	berg	#stu 130	udents

(SELECT code FROM Give UNION (SELECT code FROM Cour WHERE code (SELECT FROM	nCourses) , NULL ses NOT IN	es));			
code	period	1			
TDA357	2]	aada	NULL	1 I
TDA357	4	U	<u>code</u>	-	
TIN090	1	1	TDA100	Null	
TDA590	2	1			

F	Result	
code	period	
TDA357	2	
TDA357	4	
TIN090	1	
TDA590	2	
TDA100		

Not sets but bags!

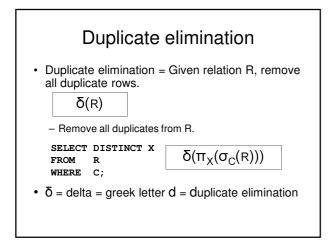
- In set theory, a set cannot contain duplicate values. Either a value is in the set, or it's not.
- In SQL, results of queries can contain the same tuples many times.
 - Done for efficiency, eliminating duplicates is costly.
- A set where duplicates may occur is called a *bag*, or *multiset*.

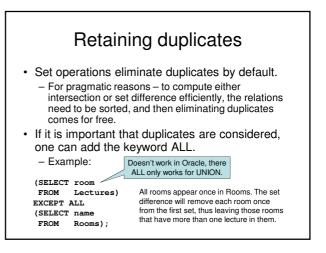


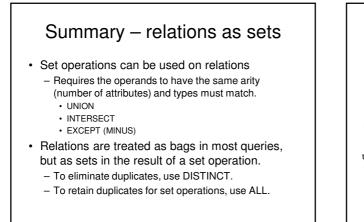
code	period		teacher		#students
TDA357	2		Niklas Brob	erg	130
TDA357	4		Rogardt He	Idal	135
TIN090	1		Devdatt Dul	ohashi	95
TDA590	2		Rogardt He	Idal	70
			ţ	SELECT FROM	teacher GivenCourse:
		teache	r	1	
		Niklas	Broberg		
		Rogard	dt Heldal		
		Devda	tt Dubhashi		
		Bogar	dt Heldal	1	

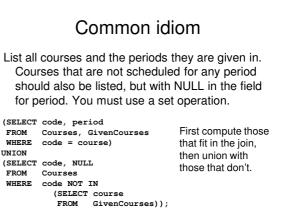
code	period		teacher		#students
TDA357	2		Niklas Brob	erg	130
TDA357	4		Rogardt He	Idal	135
TIN090	1		Devdatt Du	bhashi	95
TDA590	2		Rogardt He	Idal	70
			♥ FROM		INCT teacher nCourses;
		teacher			
		Niklas I	Broberg		
		Rogard	lt Heldal		

code	period	teacher	#students
TDA357	2	Niklas Brobe	erg 130
TDA357	4	Rogardt Hel	dal 135
TIN090	1	Devdatt Dub	hashi 95
TDA590	2	Rogardt Hel	dal 70
	UNT (teacher)	COUNT(1









Summary SQL and Relational Algebra

- SQL is based on relational algebra. - Operations over relations SELECT-FROM-
- WHERE-GROUPBY-HAVING-ORDERBY
- Operations for: - Selection of rows (σ)
 - Projection of columns (π)
 - Combining tables
 - Cartesian product (x)
 - Join, natural join, outer join (M_C, M, M)

- Grouping and aggregation Grouping (γ)
- SUM, AVG, MIN, MAX, COUNT Set operations
 - Union (U) Intersect (∩)
- Set difference (\) - Miscellaneous
 - Renaming (ρ)
 - Duplicate elimination (δ)
 - Sorting (τ)
- Subqueries
 - Sequencing
 - (Views)

Course Objectives - Usage

When the course is through, you should

- Know how to query a database for relevant data using SQL
- Know how to change the contents of a database using SQL

"Add a course 'Databases' with course code 'TDA357', given by ... "Give me all info regarding the course 'TDA357""

Exam – Relational Algebra

"Here is a schema for a database over persons and their employments. ...'

- What does this relational-algebraic expression . compute? ...
- Translate this relational-algebraic expression to SQL.
- Write a relational-algebraic expression that computes
- Translate this SQL query to a relational-algebraic expression.

Exam – SOL DML

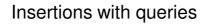
- "The grocery store wants your help in getting proper information from their database. ...
- Write a query that finds the total value of the entire • inventory of the store.
- List all products with their current price, i.e. the discount price where such exists, otherwise the base price.

Database Construction (and Usage)

More on Modifications and Table Creation Assertions Triggers

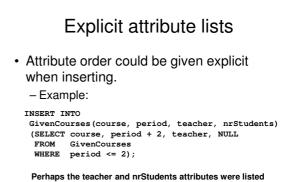
Summary – Modifications

- · Modifying the contents of a database:
 - Insertions INSERT INTO tablename VALUES tuple
 - Deletions
 - DELETE FROM tablename WHERE test over rows
 - Updates UPDATE tablename SET attribute = value WHERE test over rows



- The values to be inserted could be taken from the result of a query:
 - INSERT INTO tablename (query)
 - Example:
 - INSERT INTO GivenCourses
 (SELECT course, period + 2, teacher, NULL
 FROM GivenCourses
 WHERE period <= 2);</pre>

All courses that are given in periods one and two are also scheduled to be given two periods later, with the same teacher.



Perhaps the teacher and nrStudents attributes were listed in the other order in the definition of the table? Doesn't matter anymore since they are explicitly listed.



What will the following insertion result in?

INSERT INTO
GivenCourses(course, period, teacher)
VALUES ('TDA357', 3, 'Niklas Broberg');

 Attribute lists can be partial. Any attributes not mentioned will be given the value a default value, which by default is NULL.

Default values

- Attributes can be given default values.
 - Specified when a table is defined using the DEFAULT keyword.
 - Example: CREATE TABLE GivenCourses (course CHAR(6), period INT, teacher VARCHAR(50), nrStudents INT DEFAULT 0,

```
.. constraints ...
```

);

Default default value is NULL.

Insertion with default values Leaving out an attribute in an insertion with explicitly named attributes gives that row the default value for that attribute: INSERT INTO GivenCourses (course, period, teacher) VALUES ('TDA357', 3, 'Niklas Broberg');

 When no attribute list is given, the same effect can be achieved using the DEFAULT keyword:

INSERT INTO GivenCourses VALUES ('TDA357', 3, 'Niklas Broberg', DEFAULT);