Databases

Introduction to SQL

Nuno Antunes, Marco Vieira

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2021/2022

Outline

- Introduction to SQL
- Basic Structure of SQL Queries (select)

These slides use the following book as reference:
Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, Seventh Edition, 2019.

Structured Query Language (SQL)

- SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce after learning about the relational model, 1970's
  - Several ANSI and ISO standards over the years: 1986, 1989, 1992, 1999, 2003, 2006, 2008, 2011 and 2016
- Although called a query language, it can be used to do more tham just query a database
  - Define the structure of the database
  - Modify data
  - Specify security constraints
  - …

Consider this *Instructor* Relation...

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Basic Structure of SQL Queries (select)

- SQL DML command used to get data from the database:

```
select A1, A2,…, An
from r1, r2,…, rm
where P;
```

- A_1, A_2, …, A_n are the attributes to be obtained from the relations r_1, r_2, …, r_m
- P is a predicate to be evaluated to decide which data to consider
  - If the *where* clause is omitted, then P is true

```
select name
from instructor;
```

name
Srinivasan
Wu
Mozart
Einstein
El Said
Gold
Katz
Califieri

All Columns

```
select *
from instructor;
```

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Single Column

```
select dept_name
from instructor;
```

dept_name
Comp. Sci.
Finance
Music
Physics
History
Physics
Comp. Sci.
History
Finance
Biology
Comp. Sci.
Elec. Eng.

Distinct Rows

```
select distinct dept_name
from instructor;
```

dept_name
Finance
History
Physics
Music
Comp. Sci.
Biology
Elec. Eng.

Operations in Fields

```
select ID, name, dept_name, salary*1.1
from instructor;
```

ID	name	dept_name	?column?
10101	Srinivasan	Comp. Sci.	71500.000
12121	Wu	Finance	99000.000
15151	Mozart	Music	44000.000
22222	Einstein	Physics	104500.000
32343	El Said	History	66000.000
33456	Gold	Physics	95700.000
45565	Katz	Comp. Sci.	82500.000
58583	Califieri	History	68200.000
76543	Singh	Finance	88000.000
76766	Crick	Biology	79200.000
83821	Brandt	Comp. Sci.	101200.000
98345	Kim	Elec. Eng.	88000.000

Conditions and Ordering

```
select name
from instructor
where dept_name = 'Comp. Sci.' and salary > 70000
order by name;
```

name
Brandt
Katz

Rename Fields

```
select ID, name, dept_name, salary*1.1 as salary
from instructor
order by name;
```

ID	name	dept_name	salary
83821	Brandt	Comp. Sci.	101200.000
58583	Califieri	History	68200.000
76766	Crick	Biology	79200.000
22222	Einstein	Physics	104500.000
32343	El Said	History	66000.000
33456	Gold	Physics	95700.000
45565	Katz	Comp. Sci.	82500.000
98345	Kim	Elec. Eng.	88000.000
15151	Mozart	Music	44000.000
76543	Singh	Finance	88000.000
10101	Srinivasan	Comp. Sci.	71500.000
12121	Wu	Finance	99000.000

String Operations

```
select name
from instructor
where dept_name = 'Comp. Sci.'
order by name;
```

name
Brandt
Katz
Srinivasan

```
select name
from instructor
where dept_name = 'Comp.'
order by name;
```

(0 rows)

```
select name
from instructor
where dept_name like '%Comp.%'
order by name;
```

name
Brandt
Katz
Srinivasan

Null Values

- Null values are a difficulty, as some comparisons are not possible
  - e.g. *1 < null* *salary < 100*, if the salary can have null values

```
select *
from takes
where grade=null;
```

(0 rows)

```
select *
from takes
where grade is null;
```

id	course_id	sec_id	semester	year	grade
98988	BIO-301	1	Summer	2018	

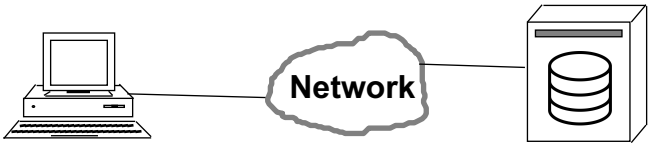
```
select *
from takes
where grade = 'A'
or grade = 'A' is unknown;
```

id	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2017	A
12345	CS-190	2	Spring	2017	A
12345	CS-315	1	Spring	2018	A
12345	CS-347	1	Fall	2017	A
76543	CS-101	1	Fall	2017	A
76543	CS-319	2	Spring	2018	A
98988	BIO-101	1	Summer	2017	A
98988	BIO-301	1	Summer	2018	

  - We can also use "*is not null*" and "*is not unknown*!"

DEMO #1

- Server: PostgreSQL server – Docker container
- Client: *psql*
- TODO:
  - Connect to database *dbcourse*, username *dblesson1*, password *dblesson1*
  - Get the data from tables *instructor* and *department*
  - Find the names of departments whose budget is greater than 82000
  - Find the IDs of all instructors that work in the finance department



psql (localhost) **PostgreSQL (Docker container)**

Q&A

Databases

Introduction to SQL

N. Antunes, Marco Vieira

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2021/2022