

Databases



IT Learning &
Outsourcing Center

Querying Multiple Tables

www.pragmatic.bg

Lector: Hristo Topuzov
E-mail: Hristo.Topuzov@pragmatic.bg

2017

Copyright © Pragmatic LLC



Short Revision

- SQL aggregate functions return a single value, calculated from values in a column
- Useful aggregate functions:
 - AVG() - returns the average value
 - COUNT() - returns the number of rows
 - MAX() - returns the largest value
 - MIN() - returns the smallest value
 - SUM() - returns the sum



Short Revision

EMPLOYEES

DEPART MENT_ID	JOB_ID	SALARY	
20	AD_ASST	4400	4400
20	MK_MAN	13000	
20	MK_MAN	12000	25000
30	PU_CLERK	2500	
30	PU_CLERK	2500	7500
30	PU_CLERK	2500	
30	PU_MAN	11000	
30	PU_MAN	11500	43500
30	PU_MAN	10000	
30	PU_MAN	11000	
...	

Grouping Data Using
Several Columns

EMPLOYEES

DPT_ID	JOB_ID	SUM(S ALARY)
20	AD_ASST	4400
20	MK_MAN	25000
30	PU_CLERK	7500
30	PU_MAN	43500
...



Short Revision

- The syntax:

SELECT <columns>, <group_function(column)>

FROM <table>

[WHERE <condition>]

[GROUP BY<group_by_expression>]

[HAVING <condition>]

[ORDER BY<columns>]

Querying Multiple Tables



- Sometimes you need data from more than one table:

LAST_NAME	DEPARTMENT_ID
King	90
Kochhar	90
Fay	20

DEPARTMENT_ID	DEPARTMENT_NAME
90	Executive
20	Marketing
10	Administration

LAST_NAME	DEPARTMENT_NAME
King	Executive
Fay	Marketing
Kochhar	Executive



Querying Multiple Tables

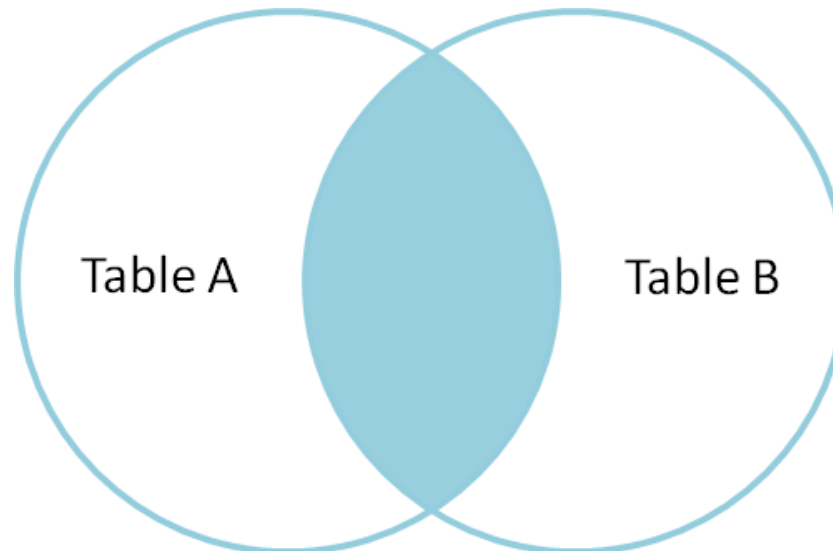


- An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them.
- The most common type of join:
 - INNER JOIN - selects all rows from both tables as long as there is a match between the columns in both tables
 - LEFT JOIN - returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.
 - FULL JOIN - returns all rows from the left table (table1) and from the right table (table2)



SQL INNER JOIN

- The syntax:
 - `SELECT column_name(s)`
`FROM table_name1 t1`
`INNER JOIN table_name2 t2`
`ON t1.column_name=t2.column_name`






SQL INNER JOIN

■ Example:

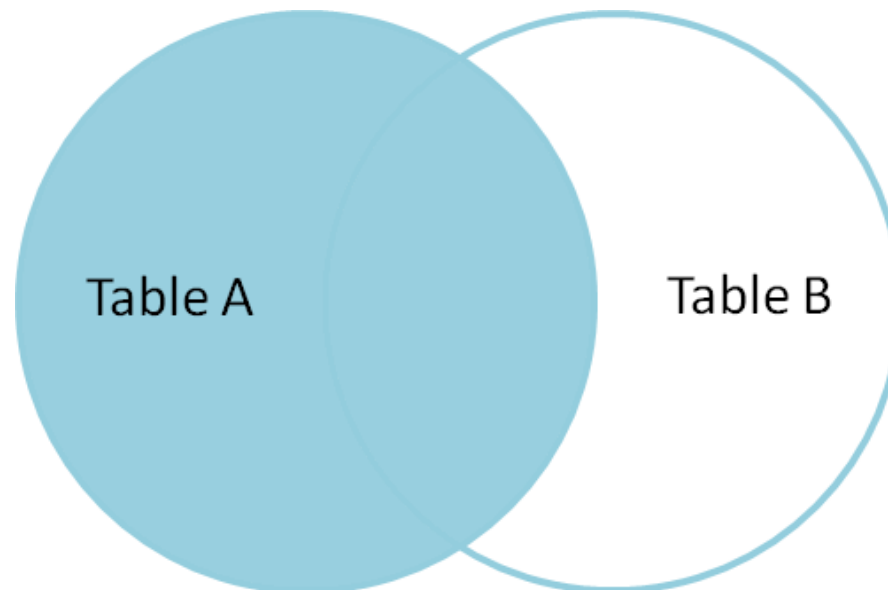
- SELECT e.name, e.email, v.status, v.fromdate, v.todate
FROM employees e
INNER JOIN vacations v on v.EmployeeId = e.id

name	 email	status	fromdate	todate
Stefan Petrov	spetrov@company.com	APPROVED	2008-07-06	2009-07-21
Jon Adams	jadams@company.com	APPROVED	2008-07-09	2009-07-19
Jon Adams	jadams@company.com	DECLINED	2008-08-14	2009-08-20
Maria Petrova	mpetrova@company.com	APPROVED	2008-06-01	2009-06-15
Maria Petrova	mpetrova@company.com	APPROVED	2009-09-09	2009-09-16
Maria Petrova	mpetrova@company.com	APPROVED	2009-11-10	2009-11-20
Dimo Ivanov	diliev@company.com	APPROVED	2010-03-20	2010-04-01
Iliana Dimitrova	idimitrova@company.com	DECLINED	2011-02-20	2011-02-27
Boris Penchev	bpenchev@company.com	DECLINED	2012-07-15	2012-07-22
Ani Yordanova	ayordanova@company.com	DECLINED	2012-08-15	2012-08-22
Ani Yordanova	ayordanova@company.com	APPROVED	2013-06-01	2013-06-15
Ivelina Borianov	iborianova@company.com	APPROVED	2013-07-10	2013-07-20
Didi Marinova	dmarinova@company.com	PENDING	2014-01-15	2014-01-22
Petia Todorova	ptodorova@company.com	PENDING	2014-02-01	2014-02-14



SQL LEFT JOIN

- The syntax :
 - SELECT column_name(s)
FROM table_name1 t1
LEFT OUTER JOIN table_name2 t2 ON
t1.column_name=t2.column_name




SQL LEFT OUTER JOIN



■ Examples:

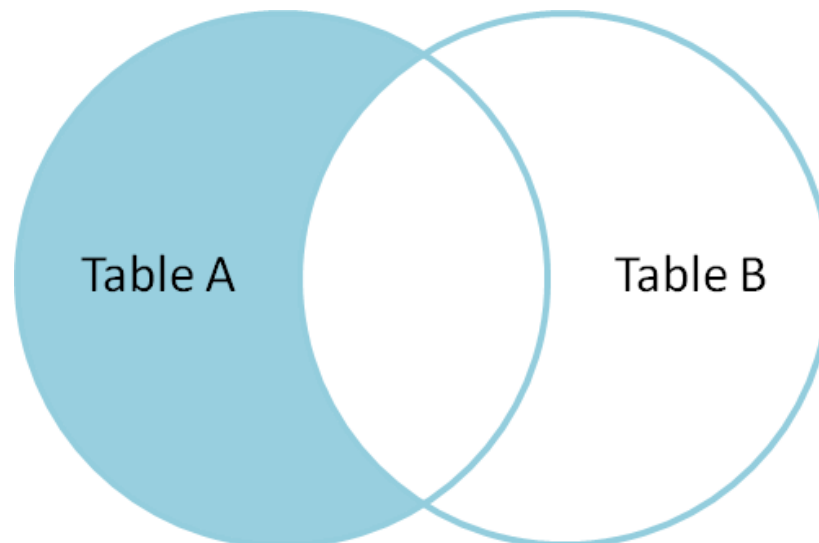
- SELECT e.name, e.email, v.status, v.fromdate, v.todate
FROM employees e
LEFT OUTER JOIN vacations v on v.EmployeeId = e.id

name	 email	status	fromdate	todate
Ronald Smith	rsmith@company.com	(NULL)	(NULL)	(NULL)
Mladen Petrov	mpetrov@company.com	(NULL)	(NULL)	(NULL)
Ivan Ivanov	iivanov@company.com	(NULL)	(NULL)	(NULL)
Todor Dimitrov	tdimitrov@company.com	(NULL)	(NULL)	(NULL)
Stefan Petrov	spetrov@company.com	APPROVED	2008-07-06	2009-07-21
Jon Adams	jadams@company.com	APPROVED	2008-07-09	2009-07-19
Jon Adams	jadams@company.com	DECLINED	2008-08-14	2009-08-20
Maria Petrova	mpetrova@company.com	APPROVED	2008-06-01	2009-06-15

SQL LEFT JOIN WHERE B.KEY IS NULL



- The syntax :
 - SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2 ON t1.column_name=t2.column_name
WHERE table_name2.column_name IS NULL




SQL LEFT JOIN WHERE B.KEY IS NULL



■ Example:

- SELECT e.name, e.email
FROM employees e
LEFT OUTER JOIN vacations v on v.EmployeeId = e.id
where v.EmployeeId is null

name	 email
Ronald Smith	rsmith@company.com
Mladen Petrov	mpetrov@company.com
Ivan Ivanov	iivanov@company.com
Todor Dimitrov	tdimitrov@company.com
Tom Blank	tblank@company.com
Teodor Jivkov	tjivkov@company.com
Veneta Petkova	vpetkova@company.com



Querying Multiple Tables

- Self join means to join a table to itself. Always used with table aliases.
- Example:
 - SELECT e.name,e.HireDate, m.name as ManagerName
FROM employees e
LEFT JOIN employees m on e.ManagerId = m.Id

name	HireDate	ManagerName
Ronald Smith	2008-03-01	(NULL)
Mladen Petrov	2008-03-01	(NULL)
Ivan Ivanov	2008-10-01	Ronald Smith
Todor Dimitrov	2009-05-12	Mladen Petrov



Querying Multiple Tables

- A three-way join is a join of three tables
- Example:
 - SELECT e.Name, e.HireDate, s.Name as Skill
FROM employees e
join employee_skills es on es.EmployeeId = e.id
join skills s on s.id = es.SkillId

Name	HireDate	Skills
Asen Gradinarov	2012-04-01	JDBC
Asen Gradinarov	2012-04-01	Java programming
Daniel Ignatov	2010-04-14	SQL programming
Daniel Ignatov	2010-04-14	Hibernate
Daniel Ignatov	2010-04-14	JDBC
Daniel Ignatov	2010-04-14	Java programming



Querying Multiple Tables

- You can apply additional conditions in the WHERE clause and sorting with ORDER BY.
- Example:
 - SELECT e.Name, s.Name as Skills
FROM employees e
join employee_skills es on es.EmployeeId = e.id
join skills s on s.id = es.SkillId
where e.ManagerId = 5
order by 1



Querying Multiple Tables

- Joins can apply any Boolean expression in the ON clause
- Example:
 - SELECT e.name as EmployeeName, d.name as
DepartmentName
FROM employees e
INNER JOIN departments d
ON e.departmentid = d.id
AND e.HireDate > adddate(curdate(),interval -3 year)

Nested SELECT Queries



- SELECT statements can be nested in the WHERE clause
- Example:

```
SELECT NAME, SALARY  
FROM EMPLOYEES  
WHERE SALARY = (SELECT MAX(SALARY)  
                FROM EMPLOYEES)
```

Note: Always prefer joins to nested SELECT statements (better performance)

Nested SELECT Queries



- SELECT statements can be nested in the FROM clause

- Example:

```
SELECT *  
FROM      (SELECT NAME, SALARY  
            FROM EMPLOYEES) AS INNER_TBL
```

Note: Always prefer joins to nested SELECT statements (better performance)



Nested SELECT Queries

- SELECT statements can be nested in the SELECT clause
- Example:

```
SELECT NAME,  
        (SELECT COUNT(1)  
         FROM EMPLOYEES A  
         WHERE A.SALARY >= B.SALARY)  
FROM EMPLOYEES B
```

Note: Always prefer joins to nested SELECT statements (better performance)



UNION Operator

- The UNION operator is used to combine the result-set of two or more SELECT statements. UNION puts lines from queries after each other.

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```



UNION Operator

- SELECT statement within the UNION must have the same number of columns
- The columns must also have similar data types
- The columns in each SELECT statement must be in the same order



UNION Operator

- Example:

- ```
SELECT id,name,departmentid
FROM employees where departmentid = 5
UNION
SELECT id,name,departmentid
FROM employees where departmentid = 6
```



# UNION Operator

- The UNION operator selects only distinct values by default. To allow duplicate values, use the ALL keyword with UNION.

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```



# UNION Operator

- Exmple:
  - SELECT id,name,departmentid  
FROM employees where departmentid = 5  
UNION ALL  
SELECT id,name,departmentid  
FROM employees where departmentid = 5



# Questions

