



## Week7\_Handout

# Database System\_SQL Introduction\_part4-6

(#)

Database System - Nankai



# Database System-SQL Introduction-part4

(#)

Database System - Nankai



# Patterns

- “Attribute **LIKE** pattern” is true if the string value of the attribute matches the pattern.
- % stands for any string
- \_ stands for any one character.

Example Drinkers(name, addr, phone)

- Find drinkers whose phone has prefix 555.

```
SELECT    name  
FROM      Drinkers  
WHERE     phone LIKE '555%' ;
```

- Also NOT LIKE for negation.

(#)

Database System - Nankai



# Patterns

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select *  
from employee  
where empname like '%风';
```

李姓员工的所有信息？

```
select *  
from employee  
where empname like '李%' ;
```

	empno	empname	deptno
▶	9031	李静	d2
	empno	empname	deptno
▶	25348	张风	d3
	28559	刘国风	d1

Database System - Nankai (#)



# NULL Values

- Tuples in SQL relations can have NULL as a value for one or more components.
- Meaning depends on context. Two common cases:
  - w *Unknown value* : e.g., we know Joe' s Bar has some address, but we don' t know what it is.
  - w *Inapplicable* : e.g., the value of attribute *spouse* for an unmarried person.

(#)

Database System - Nankai



# Comparing NULL's to Values

- The logic of conditions in SQL is really 3-valued logic: **TRUE, FALSE, UNKNOWN**.
- When any value is compared with NULL, the truth value is UNKNOWN.
- But a query only produces a tuple in the answer if its truth value for the WHERE clause is **TRUE** (not FALSE or UNKNOWN).

(#)

Database System - Nankai



# Three-Valued Logic

- To understand how AND, OR, and NOT work in 3-valued logic, think of TRUE = 1, FALSE = 0, and UNKNOWN =  $\frac{1}{2}$ .
- AND = MIN; OR = MAX,  $\text{NOT}(x) = 1-x$ .
- Example:  
TRUE AND (FALSE OR NOT(UNKNOWN))

(#)

Database System - Nankai



# Surprising Example

bar	beer	price
Joe's bar	Bud	NULL

SELECT bar  
FROM Sells  
WHERE price < 2.00 OR price >= 2.00;

UNKNOWN      UNKNOWN  
                UNKNOWN

- Joe's Bar is not produced, even though the WHERE condition is a tautology.

(#)

Database System - Nankai



# NULL Values

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

```
SELECT bar  
FROM Sells  
WHERE price IS NOT NULL;
```

```
select *  
from workson  
where job like '%';
```

	empno	projectno	job	enterdate
▶	2581	p3	分析员	1998-10-15
	9031	p1	管理员	1998-04-15
	9031	p3	职员	1997-11-15
	10102	p1	分析员	1997-01-10
	10102	p3	管理员	1999-01-01
	18316	p2	职员	1998-02-15
	28559	p2	职员	1999-02-01
	29346	p1	职员	1998-01-04

```
select projectno  
from workson  
where job is not null;
```

	projectno
▶	p3
	p1
	p3
	p1
	p3
	p2
	p2
	p1

(#)

Database System - Nankai



## 输出排序

- 对查询结果的元组按照某种顺序进行排序输出
  - ORDER BY <list of attributes>
  - ASC (升序, 默认值), DESC (降序)

```
SELECT empname  
FROM Employee  
WHERE deptno='d2'  
ORDER BY empno DESC;
```

Employee		
empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

empname
▶ 赵东生
李静
徐唱

&lt;#&gt;

Database System - Nankai



# Multi-relation Queries

- List of relations in FROM clause.
- Relation-dot-attribute disambiguates attributes from several relations with the same name.

Example Likes(drinker, beer); Frequents(drinker, bar)

Find the beers that the frequenters of Joe's Bar like.

```
SELECT  beer
FROM    Frequents, Likes
WHERE   bar = 'Joe''s Bar' AND
        Frequents.drinker = Likes.drinker;
```

(#)

Database System - Nankai



# Multi-relation Queries

Department

雇员信息以及所在部门名称和地点?

deptno	deptname	location
d1	开发部	天津
d2	财务部	北京
d3	市场部	广东

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select *
from employee, department
where
department.deptno=employee.deptno;
```

empno	empname	deptno	deptno	deptname	location
18316	冯新	d1	d1	开发部	天津
28559	刘国风	d1	d1	开发部	天津
2581	徐唱	d2	d2	财务部	北京
9031	李静	d2	d2	财务部	北京
29346	赵东生	d2	d2	财务部	北京
10102	王闻刚	d3	d3	市场部	广东
25348	张风	d3	d3	市场部	广东

```
select empno,empname,employee.deptno,deptname,location
from employee, department
where employee.deptno=department.deptno;
```

empno	empname	deptno	deptname	location
18316	冯新	d1	开发部	天津
28559	刘国风	d1	开发部	天津
2581	徐唱	d2	财务部	北京
9031	李静	d2	财务部	北京
29346	赵东生	d2	财务部	北京
10102	王闻刚	d3	市场部	广东
25348	张风	d3	市场部	广东

(#)

Database System - Nankai



# Multi-relation Queries

Department

雇员信息以及所在部门名称和地点?

deptno	deptname	location
d1	开发部	天津
d2	财务部	北京
d3	市场部	广东

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select *  
from employee natural join department;
```

	deptno	empno	empname	deptname	location
▶	d1	18316	冯新	开发部	天津
	d1	28559	刘国风	开发部	天津
	d2	2581	徐唱	财务部	北京
	d2	9031	李静	财务部	北京
	d2	29346	赵东生	财务部	北京
	d3	10102	王闻刚	市场部	广东
	d3	25348	张风	市场部	广东

(#)

Database System - Nankai



# Multi-relation Queries

Department

deptno	deptname	location
d1	开发部	天津
d2	财务部	北京
d3	市场部	广东

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

工作地在天津的所有雇员和部门信息？

```
select *  
from employee, department  
where location='天津' and  
department.deptno=employee.deptno;
```

	empno	empname	deptno	deptno	deptname	location
▶	18316	冯新	d1	d1	开发部	天津
	28559	刘国风	d1	d1	开发部	天津

```
select *  
from employee natural join department  
where location='天津';
```

	deptno	empno	empname	deptname	location
▶	d1	18316	冯新	开发部	天津
	d1	28559	刘国风	开发部	天津

(#)

Database System - Nankai



# Multi-relation Queries

Department

deptno	deptname	location
d1	开发部	天津
d2	财务部	北京
d3	市场部	广东

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

工作地在天津的雇员姓名?

```
select empname  
from employee, department  
where location='天津' and  
department.deptno=employee.deptno;
```

empname
冯新
刘国风

```
select empname  
from employee natural join department  
where location='天津';
```

(#)

Database System - Nankai



# Summary

- Patterns
- Null Values
- Order By
- Multi-relation Queries

(#)

Database System - Nankai



## 互动交流一

对于关系Student(ID,stuname,gender,birthdate),想要查询所有名字中含有“风”字的学生信息的SQL语句是

- A select \* from Student where stuname like '%风%';
- B select \* from Student where stuname like '\_风\_';
- C select \* from Student where stuname like '%风\_';
- D select \* from Student where stuname like '\_风%';

提交

(#)

Database System - Nankai



## 互动交流二

对于关系Student(ID,stuname,gender,birthdate),想要查询所有名字中第二个字为“风”字的学生信息的SQL语句是

- A select \* from Student where stuname like '%风%';
- B select \* from Student where stuname like '\_风\_';
- C select \* from Student where stuname like '%风\_';
- D select \* from Student where stuname like '\_风%';

提交

(#)

Database System - Nankai



## 互动交流三

对于关系Movies(title,year,length,genre,studioName),查询  
select \* from Movies where length<=120 or length >120;  
会返回Movies表中所有元组，这种说法

- A 正确
- B 错误
- C 不一定

提交

(#)

Database System - Nankai



## 互动交流四

对于关系Student(ID,name,score)，查询

`select * from Student where ID >= 10000 or ID < 10000`

会返回Student表中所有元组，这种说法

- A 正确
- B 错误
- C 不一定

提交

(#)

Database System - Nankai

多选题 1分



## 互动交流五一不定项选择

对于关系Student(ID,name,score)，想要查询按成绩降序排列的学生姓名，正确的SQL语句是

- A select \* from Student order by score;
- B select name from Student order by score;
- C select name from Student order by score asc;
- D select name from Student order by score desc;

提交

(#)

Database System - Nankai

主观题 10分



## 互动交流六

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

参与预算少于100000的项目的员工的工号是什么？结果按降序输出

提交

(#)

Database System - Nankai



# 讲解

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

```
select empno  
from project natural join workson  
where budget<100000  
order by empno desc;
```

empno
29346
28559
25348
18316

```
select empno  
from project, workson  
where budget<100000 and project.projectno=workson.projectno  
order by empno desc;
```

(#)

Database System - Nankai

主观题 10分



## 互动交流七

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

参与预算少于100000的项目的员工的姓名是什么？

提交

(#)

Database System - Nankai



# 讲解

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

	empname
▶	冯新
	张风
	刘国风
	赵东生

```
select empname  
from project natural join workson natural join employee  
where budget<100000;
```

(#)

Database System - Nankai



# Database System-SQL Introduction-part5

(#)

Database System - Nankai



# Explicit Tuple Variables

Sometimes we need to refer to two or more copies of a relation.

- Use *tuple variables* as aliases of the relations.

Example Beers(name, manf)

- Find pairs of beers by the same manufacturer.

```
SELECT b1.name, b2.name  
FROM Beers b1, Beers b2  
WHERE b1.manf = b2.manf AND  
      b1.name < b2.name;
```

- Why do we need ( $b1.name < b2.name$ ) ?

(#)

Database System - Nankai



# Explicit Tuple Variables

name	manf
chunsheng	A
harbin	B
miller	A
qingdao	B
snow	A

name	name
chunsheng	miller
chunsheng	snow
harbin	qingdao
miller	snow

name	name
chunsheng	chunsheng
miller	chunsheng
snow	chunsheng
harbin	harbin
qingdao	harbin
chunsheng	miller
miller	miller
snow	miller
harbin	qingdao
qingdao	qingdao
chunsheng	snow
miller	snow
snow	snow

&lt;#&gt;

Database System - Nankai



# Explicit Tuple Variables

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

如何获得隶属同一部门的员工对?

```
select e1.empname, e2.empname  
from employee e1,employee e2  
where e1.deptno=e2.deptno and  
e1.empname<e2.empname;
```

	empname	empname
▶	徐唱	李静
	张风	王闻刚
	冯新	刘国风
	徐唱	赵东生
	李静	赵东生

(#)  
Database System - Nankai



# Explicit Tuple Variables

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select e1.empname, e2.empname  
from employee e1,employee e2  
where e1.deptno=e2.deptno;
```

empname	empname
徐唱	徐唱
徐唱	李静
徐唱	赵东生
李静	徐唱
李静	李静
李静	赵东生
王闻刚	王闻刚
王闻刚	张风
冯新	冯新
冯新	刘国风
张风	王闻刚
张风	张风
刘国风	冯新
刘国风	刘国风
赵东生	徐唱
赵东生	李静
赵东生	赵东生

(#)

Database System - Nankai



# Subqueries

A query result can be used in the where-clause of another query.

Example: Sells(bar, beer, price)

- Find bars that serve Miller at the same price Joe charges for Bud.

```
SELECT bar  
FROM Sells  
WHERE beer = 'Miller' AND price =  
      (SELECT price  
       FROM Sells  
       WHERE bar = 'Joe"s Bar' AND beer = 'Bud');
```

{#}

Database System - Nankai



# Subqueries: The IN Operator

“Tuple IN relation” is true iff the tuple is in the relation.

## Example

Find the name and manufacturer of beers that Fred likes.

Beers(name, manf);Likes(drinker, beer)

```
SELECT *
  FROM Beers
 WHERE name IN
       (SELECT beer
        FROM Likes
       WHERE drinker = 'Fred' );
```

Select name,manf  
From Beers, Likes  
Where drinker= 'Fred' and  
beer=name;

Also: NOT IN.

{#}

Database System - Nankai



# Subqueries

Department

deptno	deptname	location
d1	开发部	天津
d2	财务部	北京
d3	市场部	广东

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

工作地在天津的雇员姓名？要求From子句中只有一个关系

```
select empname  
from employee  
where deptno=(select deptno from  
department where location='天津');
```

empname
冯新
刘国风

工作地在天津或北京的雇员姓名？要求From子句中只有一个关系

```
select empname  
from employee  
where deptno in (select deptno from  
department where location='天津'or  
location ='北京' );
```

empname
冯新
刘国风
徐唱
李静
赵东生

(#)

Database System - Nankai



## EXISTS

"EXISTS(relation)" is true iff the relation is nonempty.

Example: Beers(name, manf)

Find the beers that are the unique beer by their manufacturer.

```
SELECT name  
FROM Beers b1  
WHERE NOT EXISTS  
    (SELECT *  
     FROM Beers  
     WHERE manf = b1.manf  
          AND name <> b1.name);
```

(#)

Database System - Nankai



# EXISTS

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

查找部门中只有一名员工的员工姓名

```
select empname  
from employee e1  
where not exists (select *  
from employee  
where deptno=e1.deptno and  
empname<>e1.empname);
```

empname

(#)

Database System - Nankai



# The Operator ANY

- $x = \text{ANY}(\text{ <relation> })$  is true if and only if  $x$  equals **at least one** tuple in the relation.
  - That is,  $x$  is a member of the relation ( $x \text{ IN } \text{ <relation> }$ ).
- Similarly,  $=$  can be replaced by any comparison operator.
- Example:  $x > \text{ANY}(\text{ <relation> })$  means  $x$  is larger than at least one tuple in the relation.
- Equals to some
  - Note tuples must have one component only.

(#)

Database System - Nankai



# The Operator ALL

- Similarly,  $x <> \text{ALL}(\text{<relation>})$  is true if and only if for **every** tuple  $t$  in the relation,  $x$  is not equal to  $t$ .
  - That is,  $x$  is not a member of the relation ( $x \text{ NOT IN } \text{<relation>}$ ).
- The  $<>$  can be replaced by any comparison operator.
- Example:  $x \geq \text{ALL}(\text{<relation>})$  means there is no tuple larger than  $x$  in the relation.
  - Note tuples must have one component only.

(#)

Database System - Nankai



## Quantifiers – ANY, ALL

### Example

Sells(bar, beer, price)

Find the beer(s) sold for the highest price.

```
SELECT    beer
FROM      Sells
WHERE     price >= ALL(SELECT  price
                      FROM    Sells);
```

Find the beer(s) not sold for the lowest price.

```
SELECT    beer
FROM      Sells
WHERE     price > ANY(SELECT  price
                      FROM    Sells);
```

(#)

Database System - Nankai



# Quantifiers – ANY, ALL

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

选出预算最小的项目，返回项目名称

```
select projectname  
from project  
where budget<=ALL (select budget from project);
```

	projectname
▶	软件升级

(#)

Database System - Nankai



# Summary

- Explicit Tuple Variables
- Subqueries
- EXISTS
- ANY,ALL

(#)

Database System - Nankai

主观题 10分



# 互动交流一

如何获得职位相同的员工名对？

Workson

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

提交

(#)  
Database System - Nankai



# 讲解

## Workson

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select e1.empname, e2.empname  
from employee e1 natural join workson w1, employee e2 natural join workson w2  
where w1.job=w2.job and w1.empno<w2.empno;
```

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

empname	empname
徐唱	王闻刚
李静	王闻刚
李静	冯新
李静	刘国风
冯新	刘国风
李静	赵东生
冯新	赵东生
刘国风	赵东生

(#)

Database System - Nankai

主观题 10分



## 互动交流二

职位为管理员的员工姓名？要求使用子查询，from子句中只能有一个关系  
Workson

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

提交

(#)

Database System - Nankai



# 讲解

## Workson

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select empname  
from employee  
where empno in (select empno from workson where job = '管理员');
```

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

empname
李静
王闻刚

(#)

Database System - Nankai



## 互动交流三

已知关系模式  $R(a,b)$  和  $S(b,c)$ :

Q1:  $\text{SELECT } a \text{ FROM } R, S \text{ WHERE } R.b = S.b;$

Q2:  $\text{SELECT } a \text{ FROM } R \text{ WHERE } b \text{ IN } (\text{SELECT } b \text{ FROM } S);$  则

- A Q1和Q2产生的结果一样
- B Q1的结果总是包含Q2的结果
- C Q2的结果总是包含Q1的结果
- D Q1和Q2产生不同的结果

提交

(#)

Database System - Nankai



# 讲解

R		S	
a	b	b	c
A	1	1	..
B	2	1	..
C	3	2	..
D	4	2	..

Q1
A
A
B
B

Q2
A
B

Q1: SELECT a FROM R,S WHERE R.b=S.b;

Q2: SELECT a FROM R WHERE b IN  
(SELECT b FROM S);

(#)

Database System - Nankai

主观题 10分



## 互动交流四

查找只参与一个项目的员工姓名

Workson

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

提交

(#)  
Database System - Nankai



# 讲解

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select empname  
from employee  
where empno in(  
select empno  
from workson w1  
where not exists (  
select * from workson where workson.empno=w1.empno and workson.projectno <> w1.projectno));
```

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

	empname
▶	徐唱
	冯新
	张风

(#)

Database System - Nankai



## 互动交流五

Q1: `SELECT a FROM R WHERE b >= ANY (SELECT d FROM S WHERE c > 10);`

Q2: `SELECT a FROM R WHERE b >= ALL (SELECT d FROM S WHERE c > 10);` 则

- A Q1和Q2产生的结果一样
- B Q1的结果总是包含Q2的结果
- C Q2的结果总是包含Q1的结果
- D Q1和Q2产生不同的结果

提交

(#)

Database System - Nankai



# 讲解

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

```
select projectname  
from project  
where budget>=ANY (select budget from project  
where projectno='p4')
```

projectname
▶ 网络布线
▶ 软件升级
▶ 系统开发

```
select projectname  
from project  
where budget>=ALL (select budget from project  
where projectno='p4')
```

projectname
▶ 网络布线
▶ 软件升级
▶ 系统开发

(#)

Database System - Nankai



# Database System-SQL Introduction-part6

(#)

Database System - Nankai



# Union, Intersection, Difference

## Example

Likes(drinker, beer); Sells(bar, beer, price);

Frequents(drinker, bar)

- Find the drinkers and beers such that the drinker likes the beer and frequents a bar that serves it.

(SELECT \* FROM Likes)

INTERSECT

(SELECT drinker, beer  
FROM Sells, Frequents  
WHERE Frequents.bar = Sells.bar) ;

UNION, EXCEPT

{#}

Database System - Nankai



# Forcing Set/Bag Semantics

- Default for select-from-where is bag;
- Default for union, intersection, or difference is set.
- Force set semantics with DISTINCT after SELECT.
  - Expensive execution. Better be worth it.

Example: Sells(bar, beer, price)

- Find the different prices for beers.

```
SELECT DISTINCT price  
FROM   Sells;
```

(#)

Database System - Nankai



# Forcing Set/Bag Semantics

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

```
select e1.empname  
from employee e1,employee e2;
```

```
select e1.empname from employee e1,employee e2 LIMIT 0, 1000 49 row(s) returned
```

```
select distinct e1.empname  
from employee e1,employee e2;
```

empname
徐唱
李静
王闻刚
冯新
张风
刘国风
赵东生

(#)

Database System - Nankai



# Aggregations

sum, avg, min, max, count, and count(\*)

## Examples

Sells(bar, beer, price)

```
SELECT AVG(price)  
FROM Sells  
WHERE beer = 'Bud';
```

```
SELECT COUNT(DISTINCT price)  
FROM Sells  
WHERE beer = 'Bud';
```

(#)

Database System - Nankai



# Aggregations

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

所有项目的平均预算?

```
select avg(budget)  
from project;
```

	avg(budget)
▶	133533.3333

所有项目的最少预算?

```
select min(budget)  
from project;
```

	min(budget)
▶	95000

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

```
select count(empno) from workson;
```

11

```
select count(distinct empno)  
from workson;
```

7

```
select count(job) from workson;
```

8

(#)

Database System - Nankai



# Grouping

- GROUP BY (attribute list);
  - Select-from-where-group by
- The result of FROM-WHERE clauses is grouped by values of the group-by attribute list
- Aggregations take place within each group.  
Example      Sells(bar, beer, price)
- Find the average sales price for each beer.

```
SELECT    beer, AVG(price)
FROM      Sells
GROUP BY  beer;
```

(#)

Database System - Nankai



# Example

Sells(bar, beer, price);  
Frequents(drinker, bar)

- Find, for each drinker, the average price of Bud at the bars they frequent.

```
SELECT drinker, AVG(price)
FROM Frequents, Sells
WHERE beer = 'Bud' AND
      Frequents.bar = Sells.bar
GROUP BY drinker;
```

Note: grouping occurs after the  $\times$ ,  $\sigma$  operations.

(#)

Database System - Nankai



# Grouping

## Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

每个员工做几个项目?

```
select empno,count(*)  
from workson  
group by empno;
```

empno	count(*)
2581	1
9031	2
10102	2
18316	1
25348	1
28559	2
29346	2

每个项目由几名员工完成?

```
select projectno,count(*)  
from workson  
group by projectno;
```

projectno	count(*)
p1	4
p2	4
p3	3

每个职位有几名员工?

```
select job,count(*)  
from workson  
group by job;
```

job	count(*)
分析员	2
管理员	2
职员	4
NULL	3

(#)

Database System - Nankai



# Grouping

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

每个员工所参与项目的平均预算?

```
select empno,avg(budget)
from workson,project
where workson.projectno=project.projectno
group by empno;
```

```
select empno,avg(budget)
from workson natural join project
group by empno;
```

empno	avg(budget)
9031	152800.0000
10102	152800.0000
28559	107500.0000
29346	107500.0000
18316	95000.0000
25348	95000.0000
2581	185600.0000

(#)

Database System - Nankai



# Grouping

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

1998年4月15日之后入职的员工所参与项目的平均预算?

```
select empno,avg(budget)
from workson,project
where
workson.projectno=project.projectno and
enterdate>'1998-04-15'
group by empno;
```

	empno	avg(budget)
▶	2581	185600.0000
	10102	185600.0000
	25348	95000.0000
	28559	107500.0000

(#)

Database System - Nankai



# SELECT Lists With Aggregation

Sells(bar, beer, price)

- **Possible Query to find the bar that sells Bud the cheapest:**

```
SELECT    bar, MIN(price)
FROM      Sells
WHERE     beer = 'Bud';
```

- Illegal. Why?
- Rule: *Each element of a SELECT clause must either be aggregated or appear in a group-by clause.*

```
select bar
from sells
where beer= 'bud' and
      price <=ALL (select price
                    from sells
                   where beer = 'bud ');
```

**Problem:** How would we find that bar?

(#)

Database System - Nankai



# HAVING clause

- HAVING clauses are selections on groups

Example: Beers(name, manf); Sells(bar, beer, price)

- Find the average price of those beers that are either served in at least 3 bars or manufactured by Busch.

```
SELECT    beer, AVG(price)
FROM      Sells
GROUP BY    beer
HAVING    COUNT(*) >= 3      OR
          beer IN (SELECT name
                      FROM Beers
                      WHERE manf = 'Busch');
```

(#)

Database System - Nankai



# HAVING clause

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

超过2名员工参与的项目以及对应的参与员工数？

```
select projectno,count(*)  
from workson  
group by projectno  
having count(*)>2;
```

	projectno	count(*)
▶	p1	4
	p2	4
	p3	3

超过3名员工参与的项目或项目' p3' 对应项目的参与员工数。

```
select projectno,count(*)  
from workson  
group by projectno  
having count(*)>3 or projectno='p3';
```

	projectno	count(*)
▶	p1	4
	p2	4
	p3	3

(#)

Database System - Nankai



# Summary

- UNION, INTERSECT, EXCEPT
- SET/BAG SEMANTICS
- Aggregations
- Grouping
- Having

(#)

Database System - Nankai

单选题 1分



## 互动交流一

在下列查询中，R是任意的关系模式

Q1: (SELECT \* FROM R) UNION (SELECT \* FROM R);

Q2: SELECT \* FROM R; 则

- A Q1和Q2产生的结果一样
- B Q1的结果总是包含Q2的结果
- C Q2的结果总是包含Q1的结果
- D Q1和Q2产生不同的结果

提交

(#)

Database System - Nankai



# 讲解

test\_union

ID	name
1	Alex
2	Brown
3	Clark
1	Alex
2	Brown
3	Clark

(SELECT \* FROM test\_union) UNION  
(SELECT \* FROM test\_union);

SELECT \* FROM test\_union;

ID	name
1	Alex
2	Brown
3	Clark

ID	name
1	Alex
2	Brown
3	Clark
1	Alex
2	Brown
3	Clark

(#)

Database System - Nankai



## 互动交流二

Q1: `SELECT DISTINCT * FROM R;`

Q2: `(SELECT * FROM R) INTERSECT (SELECT * FROM R);` 则

- A Q1和Q2产生的结果一样;
- B Q1的结果总是包含Q2的结果;
- C Q2的结果总是包含Q1的结果;
- D Q1和Q2产生不同的结果;

提交

(#)

Database System - Nankai



# 讲解

test\_union

SELECT distinct \* FROM test\_union;

	ID	name
▶	1	Alex
	2	Brown
	3	Clark
▶	1	Alex
	2	Brown
	3	Clark

(SELECT \* FROM test\_union)  
INTERSECT  
(SELECT \* FROM test\_union);

	ID	name
▶	1	Alex
	2	Brown
	3	Clark

(#)

Database System - Nankai



## 互动交流三

Q1: `SELECT DISTINCT a FROM R WHERE b>10;`

Q2: `SELECT a FROM R WHERE b>10 GROUP BY a;` 则

- A Q1和Q2产生的结果一样
- B Q1的结果总是包含Q2的结果
- C Q2的结果总是包含Q1的结果
- D Q1和Q2产生不同的结果

提交

(#)

Database System - Nankai



# 讲解

R

SELECT DISTINCT a FROM R WHERE b>10;

	a	b
▶	1	20
	1	30
	2	40
	2	50

	a
▶	1
	2

SELECT a FROM R WHERE b>10 GROUP BY a;

(#)

Database System - Nankai



## 互动交流四

In this question, the schema is R(A, B, C).

Q1: `SELECT DISTINCT * FROM R;`

Q2: `SELECT * FROM R GROUP BY A, B, C;`

- A Q1 and Q2 produce the same answer.
- B The answer to Q1 is always contained in the answer to Q2 .
- C The answer to Q2 is always contained in the answer to Q1 .
- D Q1 and Q2 produce different answers.

提交

(#)  
Database System - Nankai



# 讲解

R\_test

SELECT distinct \* FROM R\_TEST;

	A	B	C
▶	A	B	C
A	B	C	
A	B	C	
A2	B	C	
A	B2	C	
A	B	C2	

	A	B	C
▶	A	B	C
A2	B	C	
A	B2	C	
A	B	C2	

SELECT \* FROM R\_test GROUP BY A, B, C;

(#)

Database System - Nankai

主观题 10分



## 互动交流五

属于部门d2并且参与了项目p2的员工的员工号

Workson

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

提交

(#)

Database System - Nankai



# 讲解

```
(select empno from Employee  
where deptno= 'd2' )  
  
INTERSECT  
  
(select empno from Workson  
where projectno = 'p2' );
```

MYSQL:

```
select empno  
from Employee natural join Workson  
where deptno='d2' and projectno='p2';
```

empno
29346

{#}

Database System - Nankai

主观题 10分



## 互动交流六

如果担任过职员的员工所参与的所有项目的平均预算大于十万，  
则返回结果为职员的名字(empname)，与对应的平均项目预算值，  
结果如图所示

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

	empname	avg(budget)
▶	李静	152800.0000
	刘国风	107500.0000
	赵东生	107500.0000

提交

(#)  
Database System - Nankai



# 讲解

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

```
select empname, avg(budget)
from employee natural join workson natural join project
where empno in (select empno from workson where job='职员')
group by empname
having avg(budget)>100000;
```

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

	empname	avg(budget)
▶	李静	152800.0000
	刘国风	107500.0000
	赵东生	107500.0000

(#)

Database System - Nankai



# 讲解

Employee

empno	empname	deptno
2581	徐唱	d2
9031	李静	d2
10102	王闻刚	d3
18316	冯新	d1
25348	张风	d3
28559	刘国风	d1
29346	赵东生	d2

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

```
select empname, avg(budget)
from employee natural join workson natural join project
where job='职员'
group by empname
having avg(budget)>100000;
```

?

Project

projectno	projectname	budget
p1	网络布线	120000
p2	软件升级	95000
p3	系统开发	185600

	empname	avg(budget)
▶	李静	185600.0000
◀	赵东生	120000.0000



(#)

Database System - Nankai