

# LAB 6 Notes

## The Relational Algebra

- Any questions on the project (Discuss)
- We will discuss the java program
- We will continue our discussion on SQL
- We will finish the postgres manual

Ch.1: Overview of Database Systems Ch.2:  
Introduction to Database Design Ch.3: The  
Relational Model Ch.4: Relational Algebra Ch.5:  
SQL Ch.8: Storage and Indexing

Ch.9: Storing Data: Disks and Files Ch.10:  
Tree-Structured Indexing Ch.11: Hash-  
Based Indexing Ch.12: Overview of Query  
Evaluation Ch.13: External Sorting Ch.14:  
Evaluation of Relational Operators Ch.15: A  
Typical Relational Query Optimizer Ch.16:  
Overview of Transaction Management

## Outline

**1-4) More on SQL Operators.** I will try to emphasize on the most important aspects and then jump into examples

**5) Examples on SQL.**

**6) If time permits we will look at the project manual.**

### 1) ANY, ALL

```
SELECT [DISTINCT] a
FROM from-list
WHERE attribute <=> ALL/ANY (SELECT attribute
                             FROM X
                             )
```

*Set comparison operator (union compatibility)*

“Find the oldest employee”

**We have already seen**

```
SELECT *
FROM Employee e
WHERE e.age = (SELECT MAX(age) FROM EMPLOYEE);
```

**OR**

```
SELECT *
FROM EMPLOYEE
WHERE E.age > ALL (SELECT E2.age FROM
                  EMPLOYEE E2
                  WHERE E2.ssn!=E.ssn);
```

\* **ALL => ALL in the set**

\* ANY => At least 1 in the set

ANY HERE WOULD PRODUCE: Find employees who's age is bigger than AT least somebody's else age.

## 2) GROUP BY and HAVING CLAUSE

```
SELECT [DISTINCT] a, b, c...z, SUM(A),  
FROM from-list  
WHERE qualification  
GROUP BY a, b, c...z,  
HAVING qualification-on-grouping
```

*Example: Find how many sailors belong to each group that has more than 30 members*

*Sailors(sid, name, rating, age, group)*

1	Chris	20	1
2	Chris	35	2
3	Chris	20	1
4	John	15	2

```
SELECT COUNT(id)  
FROM Sailors  
WHERE group= I i={1,2,3...}
```

```
SELECT group, count(*) as c  
FROM Sailors  
GROUP BY group, c  
HAVING c>30
```

- Everything that appears in GROUP BY is also part of the select clause

**Query: Find the age of the youngest sailor who is eligible to vote (older than 18 years) for each group with at least 2 such sailors.**

```
SELECT group, MIN(age)  
FROM Sailor  
WHERE age>18  
GROUP BY group  
HAVING COUNT(*)>1;
```

1	Chris	20	1
2	Chris	35	2
3	Chris	19	1
4	John	15	2

Until where  
result

Until group by

Until Having

1 20  
2 35  
1 19

1	20
1	19
2	35

1	19
---	----

### 3) NULLs

*unknown or inapplicable.*

Student(ssn, name, age, address)

1321, "John", null

1421, "John", 15

1521, "John", 10

1621, "John", 15

SELECT AVG(age)

FROM Student

☛  $15+10+15+0 / 4 = 10$

SELECT AVG(age)

FROM Student

WHERE age IS NOT NULL

☛  $15 + 10 + 15 / 4 = 13,33$

*Find all student that don't have their age in the system*

SELECT \* FROM Student WHERE AGE IS NULL;

### 4) Nested Queries in the FROM clause

(Not implemented in many DBMS systems)

*// give me a list of salaries (above \$20000) where each salary represents the MAX salary of some particular age.*

SELECT TEMP.salary

FROM (SELECT E.age, MAX(salary) AS salary

FROM EMPLOYEE

GROUP BY E.age

```
) AS TEMP  
WHERE TEMP.salary>2000;
```