

Note

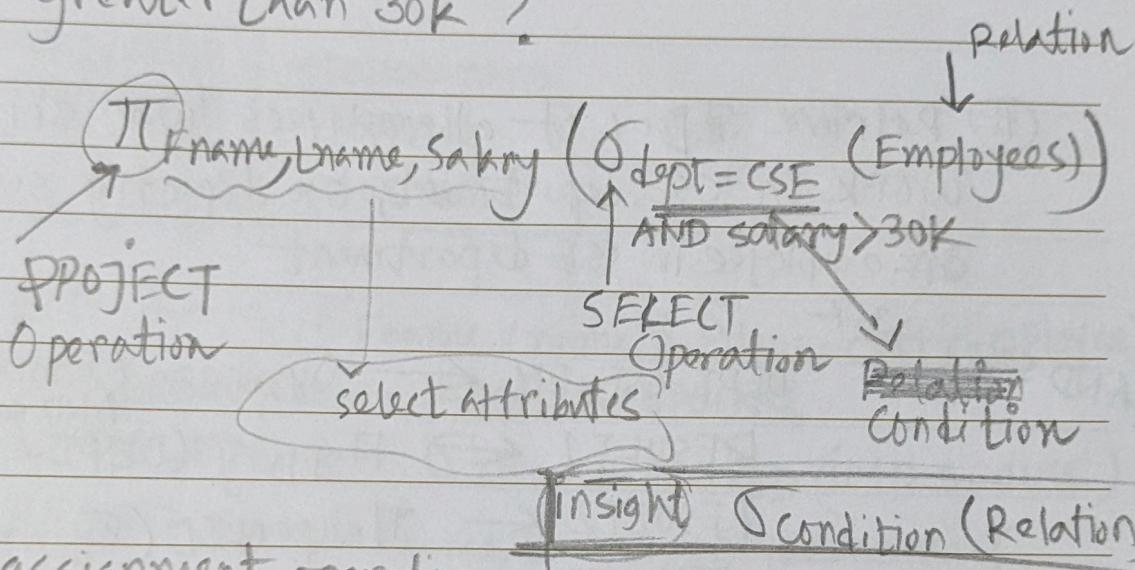
SQL (06,07)

CHAP - 08

OB - BASIC  
Date: [ ] [ ] [ ]

## # Relational Algebra

⇒ Find out Fname, Lname, salary of those who come from CSE department and salary is greater than 30k?



## # assignment operation

DEPT-SALARY ←  $\sigma_{dept = CSE \text{ AND } salary > 30k} (Employees)$

RESULT ←  $\Pi_{Fname, Lname, salary} (\text{DEPT-SALARY})$

## # RENAMING

TEMP ←  $\sigma_{dept = CSE \text{ AND } salary > 30k} (Employees)$

R(First-name, last-name, sal) ←

$\Pi_{Fname, Lname, salary} (\text{TEMP})$

Note

Date: \_\_\_\_\_

Important

Query:  
in SQL

SELECT E.Fname AS first-name,  
E.Lname AS last-name, E.salary  
FROM EMPLOYEE AS E AS sal  
WHERE E.dept=CSE AND  
E.salary > 30K

⑥ Retrieve ID of all employees who either  
work in CSE department, or directly supervise  
employees in CSE department

AND salary > 30K

DEPT-CSE-EM ← Dept=CSE (Employee)  
RESULT1 ← π<sub>ID</sub> (DEPT-CSE-EM)  
RESULT2 ← π<sub>super-ID</sub> (Dept=CSE (Employee))  
RESULT ← RESULT1 ∪ RESULT2

TABLE

ID	Fname	Lname	Salary	Dept	Supervised To
100	Abdul	Karim	35000	CSE	115
115	Abul	Hashem	31000	CSE	118
116	Abul	Kalam	30000	CSE	118

RESULT1

RESULT2

100 35000  
115 31000  
116 30000

RESULT

100
115
118

## # CARTESIAN PRODUCT

⇒ Retrieve employee name and their spouse name who works in department of CSE.

## SPOUSE

EmployeeID	spouse-name
100	Jaynob
118	Areesha

EMP ←  $\Pi_{Fname, Lname, ID} \sigma_{Dept = CSE} (Employee)$

EMP\_SPOUSE ← EMP × SPOUSE

CHOOSE\_EMP ←  $\sigma_{ID = EmployeeID} (EMP\_SPOUSE)$

RESULT ←  $\Pi_{Fname, Lname, spouse-name} (CHOOSE\_EMP)$

ID	Fname	Lname	EmployeeID	spouse-name
100	Abdul	Karim	100	Jaynob
100	Abdul	Karim	118	Areesha
115	Abul	Hashem	100	Jaynob
115	Abul	Hashem	118	Areesha

Abdul	Karim	Jaynob
-------	-------	--------

Note

Date: \_\_\_\_\_

INNER JOIN

# JOIN

EQUIJOIN

NATURAL JOIN (denoted by \*)

⇒ Rewrite PREVIOUS

Dept=CSE (E)

EMP ← π<sub>ID, Fname, Lname</sub> (Employee)

CHOOSEEMP ← EMP [ID = EmployeeID] spouse

RESULT ←

π<sub>Fname, Lname, spouse-name</sub>  
(CHOOSE\_EMP)

EQUIJOIN

⇒ Natural Join (\*)

Two join Attributes should have the same name in both relations.

In previous, we have to rename 'EmployeeID' of spouse to 'ID' if we want a natural join. OR viceversa.

EMP-SL ← π<sub>ID, Fname, Lname</sub> (Employee)

EMP ← P (EmployeeID, Fname, Lname) (EMP-SL)

CHOOSEEMP ← EMP \* spouse

OR,

CHOOSEEMP ← EMP \* P (ID, spouse-name)  
Renamed  
Spouse

Rename Operator

RESULTS SHOULD BE

ID	Fname	Lname	Spouse-name
100	Abdul	Karim	Jaynob

SQL FOR EQUIJOIN

```
SELECT E.Fname, E.Lname, S.Spouse-name
FROM Employee AS E JOIN Spouse AS S
ON E.ID = S.EmployeeID
WHERE E.dept = CSE;
```

(OR NESTED)

```
SELECT E.Fname, E.Lname, E.ID
FROM Employee AS E
WHERE E.dept = CSE IN (
```

```
SELECT S.Spouse-name
FROM Spouse AS S
WHERE S.EmployeeID = E.ID
```

(#(NATURAL JOIN))

```
SELECT Fname, Lname, Spouse-name
FROM Employee NATURAL JOIN (SELECT EmployeeID AS ID
                           FROM Spouse) AS NEW-SPOUSE
WHERE dept = CSE;
```

## INNER JOIN

TYPE OF match - and - combine operation  
 combination of 'CARTESIAN PRODUCT'  
 AND 'SELECTION'

## # AGGREGATE FUNCTION

⇒ Find out total Employees, and their average salary of EACH department.

⟨grouping attributes⟩  $\sum$  ⟨Function list⟩  $(R)$

- SUM
- AVERAGE
- MAXIMUM
- MINIMUM
- COUNT

$P$   $R(\text{dept}, \text{Total Employees}, \text{Average Salary})$   $\sum_{\text{dept}}$  COUNT ID, AVERAGE salary (Employee)

dept	TOTAL Employees	Average Salary
CSE	3	31000
EEE	1	33000

⇒ Grouping is necessary, now the functions are applied to all the tuples in the relation, so the resulting relation has a single tuple only.

⇒ Null value isn't considered here to aggregate.

## # SQL

COUNT(ID)

```
SELECT dept, COUNT(*) AS TotalEmployee,
FROM Employee AS E AVG(Salary) AS AverageSalary
GROUP BY dept;
```

## # LEFT/RIGHT OUTER JOIN / FULL OUTER

### # Left Outer Join

ID	Fname	Lname	Salary	dept	supervised To
100	Abdul	Karim	35000	CSE	115
115	Abul	Hashem	31000	CSE	118
116	Abul	Kalam	30000	CSE	118
950	RAHUL	KAUL	32000	EEE	118

Spouse

EmployeeID	Spouse Name
100	Jaynob
115	Aesha
116	Jui

Q for left outer join:

keeps every tuple in the first/left Relation

R  $\bowtie$  S# find out spouse name of every employee.

(LEFT)

TEMP  $\leftarrow$  EMPLOYEE  $\bowtie$  ID=EmployeeID SpouseRESULT  $\leftarrow$  TT Fname, Lname, Spouse\_name (TEMP)

ID	Fname	Lname	Spouse_name
100	Abdul	Karim	Jaynob
115	Abul	Hashem	NULL
116	Abul	Kalam	Aesha
950	Rahul	Kaul	NULL

Right  
Relation  
will be filled  
with NULL  
if mismatched.

# Q for Right outer Joinkeeps every tuple in the second,  
in first filled with NULL if  
mismatched.

⇒ Find out husband name of every spouse.

**RIGHT**

TEMP ← SPOUSE  $\bowtie$  EMPLOYEE  
ID = EmployeeID

RESULT ←  $\Pi$  spouse-name, Fname, Lname (TEMP)

R( Name, Husband's Fname, Husband's Lname )

ID	Name	Fname	Lname
100	Jaynab	Abdul	Karim
116	Aesha	Abul	Kalam
111	Jui	NULL	NULL

**#**

FULL OUTER JOIN  $\bowtie$

keeps all tuples in both the left and right relations when no matching tuples are found.

**#** Find out spouse name of every employee using outer JOIN

**FULL**

TEMP ← EMPLOYEE  $\bowtie$  <sup>spouse</sup> ID = EmployeeID

RESULT ←  $\Pi$  ID, spouse-name (TEMP)

ID	spouse-name
100	Jaynab
115	NULL
116	Aesha
950	NULL
NULL	Jui

Note

Date: \_\_\_\_\_

LEFT

SELECT E.Fname, E.Lname, S.spouse-name  
FROM EMPLOYEE AS E **LEFT JOIN**  
SPOUSE AS S ON E.ID = S.  
EmployeeID

FULL

FULL OUTER JOIN

RIGHT

RIGHT JOIN

## # ADVANCED SQL

(Follow company database schema)

Q1: Retrieve the names of all employees who don't have supervisors.

SELECT Fname, Lname

FROM employee

WHERE super-ssn IS NULL;

Q2: Find out the project numbers of projects that have an employee with last name 'Smith' involved as manager. FOR involved as worker.

(S2 #1)

SELECT Pnumber ~~DISTINCT~~ ~~Noneed~~

FROM PROJECT JOIN DEPARTMENT ON Dnumber

WHERE Pnumber IN (SELECT ~~Pnumber~~ = Dnum

WHERE Mgr-ssn IN (SELECT ssn

FROM Employee

WHERE ssn = Mgr-ssn AND  
Lname = ('Smith'))

Note

Date: \_\_\_\_\_

Q2: Find out Fname, Lname of all managers.

SELECT Fname, Lname  
FROM EMPLOYEE  
WHERE SSN IN (SELECT Mgr-SSN  
FROM DEPARTMENT  
WHERE SSN = Mgr-SSN);

(OR)

SELECT Fname, Lname  
FROM EMPLOYEE, DEPARTMENT  
WHERE SSN = Mgr-SSN;

(OR)

SELECT Fname, Lname  
FROM EMPLOYEE JOIN DEPARTMENT ON  
SSN = Mgr-SSN;

Q3 ANSWER

Q3

SELECT DISTINCT Pnumber

FROM PROJECTS

WHERE Pnumber IN (SELECT

FROM PROJECT, DEPARTMENT), EMPLOYEE

WHERE Dnum = Dnumber AND

FIRST

SSN = Mgr-SSN AND Lname = 'smith';

Q3

SELECT Pnumber

FROM (PROJECT JOIN DEPARTMENT ON Dnumber = Dnum)

JOIN EMPLOYEE ON Mgr-SSN = SSN

WHERE Lname = 'smith';

Date: [ ] [ ] [ ]

te

(Q4) Find out Social Security Number of all employees who work the same (project name, hours) combination on some project where employee 'John Smith' works on.

if not

```
SELECT ESSN
FROM WORKS-ON JOIN PROJECT ON Pno =
WHERE (Pname, Hours) IN
      ( SELECT (Pname, Hours)
        FROM (EMPLOYEE JOIN WORKS-ON
              ON ESSN=SSN) JOIN
              PROJECT ON Pno=Pnumber
        WHERE Fname='John' AND
              Lname='Smith' );
```

```
SELECT ESSN
FROM WORKS-ON
WHERE (Pno, Hours) IN
      ( SELECT (Pno, Hours)
        FROM WORKS-ON JOIN EMPLOYEE
              ON ESSN=SSN
        WHERE Fname='John' AND
              Lname='Smith' );
```

Note

Date: [ ] [ ] [ ]

(Q5) Names of employees whose salary is greater than the salary of all employees in department No 5?

```
SELECT Fname, Lname  
FROM EMPLOYEE  
WHERE salary > ALL ( SELECT salary  
                      FROM EMPLOYEE  
                     WHERE Dno=5 ) ;
```

OR

```
SELECT Fname, Lname  
FROM EMPLOYEE  
WHERE salary > ( SELECT MAX(salary)  
                  FROM EMPLOYEE  
                 WHERE Dno=5 ) ;
```

Note

Date: \_\_\_\_\_

(Q6) Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee.

MUST BE ALIAS

```
SELECT E.Fname, E.Lname  
FROM DEPENDENT JOIN EMPLOYEE ON  
AS D E.SSN = E.SSN AS E  
WHERE E.Fname = D.Dependent-name AND  
E.Sex = D.Sex;
```

OR

```
SELECT E.Fname, E.Lname  
FROM EMPLOYEE AS E  
WHERE E.SSN IN  
( SELECT D.ESSN  
FROM DEPENDENT AS D  
WHERE E.Fname = D.Dependent-name  
AND E.Sex = D.Sex );
```

OR

```
FROM EMPLOYEE AS E, DEPENDENT  
AS D
```

mysql -u root -p

- # For each department, retrieve the department number, sum of the distinct salaries of all employees, the maximum salary, minimum salary, average salary, and all employees number.

```

SELECT Dno, SUM(DISTINCT salary) AS TOTAL-SALARY,
       MAX(salary) AS MAX-SALARY, MIN(salary),
       AVG(salary) AS AVERAGE-SALARY,
       COUNT(*) AS TOTAL-EMPLOYEES
  FROM EMPLOYEE
 WHERE GROUP BY Dno ;
  
```

- # Find the sum of all salaries of employees of 'Research' department.

```

SELECT SUM(salary) AS TOTAL-SALARY
  FROM EMPLOYEE JOIN DEPARTMENT ON
        Dno = Dnumber
 WHERE Dname = 'Research' ;
  
```

- # Retrieve name of those employees who has more than 2 dependents.

```

SELECT Fname, Lname
  FROM EMPLOYEE
 WHERE (SELECT count(*)
        FROM DEPENDENT
       WHERE Ssn = Essn) > 2 ;
  
```

- ④ for each project on which more than two employees work, retrieve the project number and the number of employees who work on the project.

```
SELECT Pnumber, COUNT(*)  
FROM PROJECT, WORKS_ON  
WHERE Pnumber = Pno  
GROUP BY Pnumber  
HAVING COUNT(*) > 2
```

- ⑤ for each department that has more than five employees, retrieve the department number and the number of its employees who are making more than \$40,000.

```
SELECT Dno, COUNT(*)  
FROM EMPLOYEE  
WHERE salary > 40,000 AND Dno IN  
( SELECT Dno  
FROM EMPLOYEE  
GROUP BY Dno  
HAVING COUNT(*) > 5 )  
GROUP BY Dno;
```