ASSIGNMENT # 02 + 03

NAME: ABDUL-AHAD TARIQ
ROLL NO# 22L-7866 SE-4A

Write SQL Queries and Relational Algebra for the following scenarios.

QUESTION #01

1- List the part number of every part that is shipped by more than one supplier.

⇒ Select Distinct pnum from Shipments
group by pnum
Having COUNT (Distinct snum) >1;

⇒ $T1(pnum, No\_of\_suppliers) \leftarrow {}_{pnum}\mathcal{F}_{COUNT\ snum}(Shipments)$

$T2 \leftarrow \sigma_{No\_of\_Suppliers > 1}(\delta_{snum}(T1))$

$RESULT \leftarrow \pi_{pnum}(\delta_{pnum}(T2))$

2- Find the average weight of all parts.

⇒
```
SELECT AVG(weight) AS avg_weight
FROM Parts;
```

⇒ $\rho_{RESULT(avg\_weight)}\left(\mathcal{F}_{AVERAGE\ weight}(Parts)\right)$

3- For each part, list the part number and the total quantity in which that part is shipped and order the results in descending order of the total quantity shipped. Name the total quantity shipped in the result as total Shipped.

⇒
```
SELECT pnum, SUM(QTY) AS Total_Shipped
FROM Shipments
GROUP BY pnum
ORDER BY total_Shipped DESC;
```

⇒ $\pi_{Pnum,total\_shipped}\left(\tau_{SUM\ QTY\ AS\ Total\_shipped}\left({}_{pnum}\mathcal{F}_{SUM\ QTY}(Shipments)\right)\right)$

4- List only the names of those suppliers who ship a part that weighs more than 200.

⇒
```
SELECT DISTINCT SName
FROM ~~[illegible]~~ Suppliers S
INNER JOIN Shipments SH ON S.SNum=SH.SNum
INNER JOIN Parts P ON SH.PNum = P.PNum
WHERE P.weight > 200;
```

⇒ $\pi_{SNAME}(\sigma_{P.weight > 200}(Suppliers(S) \bowtie_{S.snum = Sh.snum} (Shipments(SH)$

$\bowtie_{Sh.pnum = P.pnum} Parts(P))))$

5- List the name of those cities in which both a supplier and a job are located.

⇒
```
SELECT DISTINCT S.City
FROM Suppliers S
INNER JOIN Jobs j ON S.City = j.City;
```

⇒ $\pi_{City}(\sigma_{Suppliers.City = Jobs.City}(Suppliers \times Jobs))$

Scanned with CamScanner

6- List the names of those jobs that receive a shipment from Supplier number S1.

⇒
```
SELECT J.JName
 FROM Jobs J
INNER JOIN Shipments S on J.JName = S.JName
 WHERE S.SNum = 'S1';
```

⇒ $\pi_{JName}(\sigma_{SNUM='S1'}(Jobs\ J \bowtie_{J.JName=S.JName} Shipments\ S))$

7- List the names of those Parts that are not shipped to any job

⇒
```
SELECT PName
FROM Parts P
LEFT JOIN Shipments S ON P.PNum = S.PNum
WHERE S.PNum IS NULL;
```

⇒ $T1 \leftarrow \pi_{pn\overset{u}{\cancel{a}}me}(Parts)$

$T2 \leftarrow \pi_{pnum}(Shipments)$

$RESULT \leftarrow (T1 - T2)$

8- List the names of those suppliers who ship part number P2 to any job

⇒
```
SELECT S.SName
FROM Suppliers S
INNER JOIN Shipments SH ON S.SNum = SH.SNum
WHERE SH.PNum = 'P2';
```

⇒ $\pi_{sname}(\sigma_{pnum='P2'}(\text{Suppliers } s \bowtie_{s.snum=sh..num} \text{Shipments } sh))$

9- List the names of those suppliers who ship part at least one red part to any job.

⇒
```
SELECT S.SNAME FROM Suppliers S
INNER JOIN Shipments SH ON S.SNum = SH.SNum
INNER JOIN Parts P ON SH.PNum = P.PNum
WHERE P.Color = 'RED';
```

⇒ $\pi_{sname}(\sigma_{color='RED'}(\text{Suppliers } s \bowtie_{s.snum=sh.snum} (\text{Shipments } sh \bowtie_{sh.pnum=p.pnum} \text{Parts} p)))$

10- List the part number for every part that is shipped more than once.

⇒ SELECT DISTINCT Pnum
FROM Shipments GROUP BY Pnum
HAVING COUNT(*) >1;

⇒ $T1(\text{Pnum}, \text{Shipped\_count}) \leftarrow {}_{\text{pnum}}\mathfrak{F}_{\text{COUNT}(*)}(\text{Shipments})$

$T2 \leftarrow \sigma_{\text{Shipped\_count}>1}(T1)$

$\text{RESULT} \leftarrow \pi_{\text{Pnum}}(\delta_{\text{Pnum}}(T2))$

QUESTION #02

1- find the name (first_name, last_name) and the salary of the employees who have a higher salary than the employee whose last_name = 'Bull'

⇒ SELECT (first_name + ' ' + Last_name)
as Name, Salary
FROM Employee WHERE Salary >(
SELECT Salary FROM Employee
WHERE last_name = 'Bull');

$$T1 \leftarrow \sigma_{salary>(\pi_{salary}(\sigma_{last\_name='Bull'}(Employee)))}(Employee)$$

$$\pi_{first\_name||last\_name,salary}(T1)$$

2- find the name of all employees who works in the IT department.

```
⇒ SELECT (first_name +' '+last_name) as Name
  FROM Employee e
  WHERE DEPARTMENT_ID =(
  SELECT DEPARTMENT_ID FROM
   Departments d
  WHERE d.DEPARTMENT_NAME ='IT');
```

$$\Rightarrow \pi_{first\_name||last\_name}(\sigma_{DEPARTMENT\_ID=(\pi_{DEPARTMENT\_ID}(\sigma_{DEPARTMENT\_NAME='IT'}(Departments)))}(Employee))$$

3- find the name of the employees who have a manager and worked in a USA based department

```
⇒ SELECT (e.first_name+' '+e.last_name) as Name
  FROM Employee e
  WHERE e.manager_id IS NOT NULL
  AND e.department_id IN (
SELECT department_id FROM
departments d JOIN Locations l
ON d.location_id = l.location_id
WHERE l.country_id = 'US');
```

⇒ $\pi_{first\_name, last\_name}(\sigma_{manager\_id \neq NULL \wedge department\_id \in (}$ $(Employee)$

$\pi_{department\_id}(Departments\ d \bowtie_{d.location\_id = location\_id} (\sigma_{country\_id = 'US'}(Locations))))$

4. **find those employees who earn more than the average salary. Return employee ID, first name, last name**

⇒

```
SELECT employee_id, first_name, last_name
FROM Employee WHERE salary > (
SELECT AVG(Salary) FROM Employee);
```

⇒ $\pi_{employee\_id, first\_name, last\_name}(\sigma_{salary > (\mathcal{F}_{AVG(salary)}(Employee))}(Employee))$

5- find those employees whose department is located at 'Toronto'. Return first name, last name, employee ID, job ID

⇒
```
SELECT e.first_name, e.last_name,
e.employee_id, e.job_id FROM Employee e
JOIN Departments d ON e.department_id =
d.department_id JOIN Locations l
ON d.location_id = l.location_id
WHERE l.city = 'Toronto';
```

⇒ $\pi_{first\_name, last\_name, employee\_id, job\_id}\left(Employee\ e \bowtie_{e.department\_id = department\_id} Departments\ d \bowtie_{d.location\_id = location\_id} \left(\sigma_{city='Toronto'}(Locations)\right)\right)$

6- find those employees who report to that manager whose first name is 'Payam'. Return first name, last name, employee ID and salary.

⇒
```
SELECT e.first_name, e.last_name,
e.employee_id , e.salary FROM Employee e
WHERE e.manager_id = (
 SELECT employee_id FROM Employee
 WHERE first_name = 'Payam');
```

⇒ $\pi_{first\_name, last\_name, employee\_id, salary}(\sigma_{manager\_id=}$ $(\pi_{employee\_id}(\sigma_{first\_name='Payam'}(Employee)))(Employee))$

7- find all those departments where at least one employee is employeed. Return department name.

⇒
```
SELECT department_name
FROM Departments d WHERE EXISTS (
SELECT 1 FROM Employee e
WHERE e.department_id = d.department_id);
```

⇒ $\pi_{department\_name}(Departments\ d \bowtie_{d.department\_id = department\_id} Employee)$

8- find those employees who do not work in the departments where managers' IDs are between 100 and 200. Return all the fields of the employees.

⇒ SELECT * FROM Employee e
WHERE e.department_id NOT IN(
SELECT department_id FROM Departments d
WHERE d.manager_id BETWEEN 100 AND 200);

⇒ $$\sigma_{department\_id \notin}(\pi_{department\_id}(\sigma_{manager\_id\ BETWEEN\ 100\ AND\ 200}(Departments)))(Employee)$$

9- Find those employees whose salary matches the lowest salary of any of the departments. Return first name, last name and department ID.

⇒ SELECT e.first_name, e.last_name,
e.department_id FROM Employee e
where e.salary = (
SELECT MIN(salary) FROM Employee e1
GROUP BY e1.department_id);

⇒ $$\pi_{first\_name, last\_name, department\_id}(\sigma_{salary = (_{department\_id}\Im_{MIN(Salary)}(Employee))}(Employee))$$

10- find the name of the employees who are managers.

⇒
```
SELECT (first_name + ' ' + last_name) As Name
FROM Employee e WHERE EXISTS (
SELECT 1 FROM Employee e1
WHERE e1.manager_id = e.employee_id);
```

⇒ $\pi_{first\_name || last\_name}(\sigma_{EXISTS(\pi_1(Employee \bowtie_{manager\_id = employee\_id} Employee))}(Employee))$

11- find those employees whose Salary is lower than that of employees whose job id is "MK_MAN". Exclude employees of job id 'MK_MAN'. Return employee ID, first name, last name, job ID.

⇒
```
SELECT employee_id, first_name,
last_name, job_id FROM Employee e
WHERE e.job_id <> 'MK_MAN' AND
e.salary < (
SELECT MIN(Salary) FROM Employee e1
WHERE e1.job_id = 'MK_MAN);
```

⟹ $\pi_{employee\_id, first\_name, last\_name, job\_id} (\sigma_{job\_id \neq 'MK\_MAN'}$ AND $salary < (\sigma_{job\_id = 'MK\_MAN} (\mathcal{F}_{MIN(salary)} (Employee))) (Employee))$

12- Find the name and Salary of the employees whose Salary is greater than the average Salary.

⟹
```
SELECT (first_name+' '+last_name) As Name,
salary FROM Employee WHERE salary > (
SELECT AVG(salary) FROM Employee);
```

⟹ $\pi_{first\_name || ' ' || last\_name, salary} (\sigma_{salary > (\mathcal{F}_{AVG(salary)} (Employee))} (Employee))$

13- Find the name and Salary of the employees whose Salary is equal to the minimum salary for their job grade

⟹
```
SELECT (e.first_name+' '+e.last_name)
As Name, e.salary FROM Employee e
JOIN Jobs j ON e.job_id = j.job_id
WHERE e.salary = j.min_salary;
```

⇒ $\pi_{first\_name, last\_name, salary}(\sigma_{salary=min\_salary}($ Employee e $\bowtie_{e.job\_id=job\_id}$ Jobs$))$

14- Find the name and salary of the employees who earns more than the average salary and works in any of the IT department.

⇒
```
SELECT (e.first_name+'  '+e.last_name)
As Name, e.salary FROM Employee e
JOIN Departments d ON e.department_id=
d.department_id WHERE e.salary>(
SELECT AVG(Salary) FROM Employee)
AND d.department_name = 'IT'
```

⇒ $\pi_{first\_name||'||last\_name, salary}(\sigma_{salary>(\mathcal{F}_{AVG\ salary}(Employee)) AND\ department\_name='IT'}($ Employee e $\bowtie_{e.department\_id=department\_id}$ Departments$))$

15- Find the name and salary of the employees who earns more than the earning of Mr. Bell.

⇒
```
SELECT (first_name + ' ' + last_name) As Name,
Salary FROM Employee WHERE salary >(
SELECT Salary FROM Employee
WHERE first_name = 'Bell');
```

⇒ $\pi_{first\_name \| '' \| last\_name, salary}(\sigma_{salary > (\sigma_{last\_name = 'Bull'}(Employee))}(Employee))$

16- Find the name and salary of the employees who earn the same salary as the minimum salary for all departments.

⇒
```
SELECT (e.first_name + ' ' +e.last_name) As Name,
e. salary FROM Employee e WHERE
e.salary = (
SELECT MIN (salary) FROM Employee e1
GROUP BY e1. department_id);
```

Scanned with CamScanner

⇒ $\pi_{first\_name||' '||last\_name,\ salary}\left(\sigma_{salary=(department\_id\ \mathcal{G}_{MIN\ salary}(Employee))}(Employee)\right)$

17- Find the name and salary of the employees whose salary is greater than the average salary of all departments

⇒
```
SELECT (e.first_name+' '+e.last_name)
As Name, e.salary FROM Employee e
WHERE e.salary > (
  SELECT AVG(salary) FROM Employee e1
GROUP BY e1.department_id);
```

⇒ $\pi_{first\_name||' '||last\_name,\ salary}\left(\sigma_{salary>(department\_id\ \mathcal{G}_{AVERAGE\ salary}(Employee))}(Employee)\right)$

18- Find the 3rd maximum salary in the employees table.

⇒
```
SELECT salary FROM (
  SELECT salary, DENSE_RANK() OVER
(ORDER BY salary DESC) AS rant
  FROM Employee) AS ranked_salaries
WHERE rank=3;
```

⇒ $\pi_{Salary}\left(\sigma_{row\_num=3}\left(\pi_{salary,\ DENSE\_RANK()\ OVER(\tau_{salary})\ AS\ row\_num}(Employee)\right)\right)$