

ASSIGNMENT1:

Part1:

1. Sql Query:

select student.student_id, student.name from student inner join guide on student.phone=guide.student_phone
group by student.student_id, student.name having count(guide.guide_id)>1

Relational Algebra:

$\pi_{\text{student.student_id, student.name}} \sigma_{\text{COUNT (guide_id) > 1}} \gamma_{\text{student_id, name, COUNT (guide_id)}} (\text{student} \bowtie_{\text{student . phone = guide . student_phone}} \text{guide})$

2. Sql Query:

select s1.name, s1.dept from student s1 inner join (select avg(gpa) avg_gpa, dept from student group by dept) s2
on s1.dept=s2.dept where s1.gpa>s2.avg_gpa

Relational Algebra: $\rho \rightarrow$ Rename

$\pi_{s1.name, s1.dept} \sigma_{s1.gpa > s2.avg_gpa} (\rho_{s1} \text{ student} \bowtie_{s1.dept = s2.dept} \rho_{s2} \pi_{AVG(gpa) \rightarrow avg_gpa, dept} \gamma_{dept, AVG(gpa)} \text{ student})$

3. Sql Query:

select name from student s where gpa=(select max(gpa) from student where dept=s.dept)

Relational Algebra:

$\pi_{\text{name}} (\text{student} \bowtie_{\text{gpa=max_gpa}} \gamma_{\text{dept, max(gpa) \rightarrow max_gpa}} (\text{student}))$

4. Sql Query:

select s1.name as student1, s1.dept as dept1, s2.name as student2, s2.dept as dept2, g1.guide_name from
student s1 join guide g1 on s1.phone = g1.student_phone join student s2 on s1.phone != s2.phone join guide g2
on s2.phone = g2.student_phone and g1.guide_id = g2.guide_id where s1.dept != s2.dept

Relational Algebra:

$\pi_{s1.name \rightarrow student1, s1.dept \rightarrow dept1, s2.name \rightarrow student2, s2.dept \rightarrow dept2, g1.guide_name} (\sigma_{s1.dept \neq s2.dept} (\rho_{g1}(\text{guide}) \bowtie_{s1.phone = g1.student_phone} \rho_{s1}(\text{student}) \bowtie_{s1.phone \neq s2.phone} \rho_{s2}(\text{student}) \bowtie_{s2.phone = g2.student_phone \wedge g1.guide_id = g2.guide_id} \rho_{g2}(\text{guide})))$

5. Sql Query:

select g.guide_name, g.guide_id, avg(s.gpa) as avg_gpa from guide g join student s on g.student_phone=s.phone
group by g.guide_name, g.guide_id order by avg_gpa desc limit 1;

Relational Algebra: $\rho \rightarrow$ Rename, $\tau \rightarrow$ retrieve top records

$\tau_{avg_gpa} \pi_{g.guide_name, g.guide_id, AVG(gpa) \rightarrow avg_gpa} \gamma_{guide_name, guide_id, AVG(gpa)} (\rho_g \text{ guide} \bowtie_{g.student_phone = s.phone} \rho_s \text{ student})$

6. Sql Query:

select student.student_id, student.name, student.phone from student inner join guide on
student.phone=guide.student_phone

Relational Algebra:

$\pi_{\text{student . student_id, student . name, student . phone}} (\text{student} \bowtie_{\text{student . phone = guide . student_phone}} \text{guide})$

7. Sql Query:

select g.guide_name from guide g join student s on g.student_phone=s.phone group by g.guide_name having
count(distinct s.dept) > 1

Relational Algebra: $\rho \rightarrow$ Rename

$\pi_{g.guide_name} \sigma_{\text{COUNT (distinct dept) > 1}} \gamma_{guide_name, COUNT (dept)} (\rho_g \text{ guide} \bowtie_{g.student_phone = s.phone} \rho_s \text{ student})$

8. Sql Query:

select s.name from student s where s.dept in (select distinct s2.dept from guide g join student s2 on
g.student_phone= s2.phone where g.guide_name = 'Dr. Jennifer') group by s.name having count(distinct s.dept)

= (select count(distinct s3.dept) from guide g2 join student s3 on g2.student_phone= s3.phone where g2.guide_name = 'Dr. Jennifer')

Relational Algebra:

Sub Query1: $S1 \leftarrow \pi_{s2.dept}(\sigma_{g.guide_name='Dr.Jennifer'}(guide) \bowtie_{g.student_phone=s2.phone} student)$

Sub Query2: $S2 \leftarrow \gamma_{count(distinct s3.dept) \rightarrow count_dept}(\sigma_{g2.guide_name='Dr.Jennifer'}(guide) \bowtie_{g2.student_phone=s3.phone} student)$

Complete Query: $\pi_{s.name}(\sigma_{dept_count=S1.count_dept}(\gamma_{s.name, count(distinct s.dept) \rightarrow dept_count}(student)) \bowtie_{s.dept \in S2} S2)$

9. Sql Query:

select g.guide_name from guide g join student s on g.student_phone= s.phone where s.dept = 'CSE' group by g.guide_name having count(*) = (select max(guide_count) from (select count(*) as guide_count from guide g2 join student s2 on g2.student_phone= s2.phone where s2.dept = 'CSE' group by g2.guide_name) as guide_counts)

Relational Algebra:

Sub Query1: $S1 \leftarrow \gamma_{g2.guide_name, count(*) \rightarrow guide_count}(\sigma_{s2.dept='CSE'}(guide \bowtie_{g2.student_phone=s2.phone} student \ s2))$

Sub Query2: $S2 \leftarrow \gamma_{max(guide_count) \rightarrow max_guide_count}(S1)$

Complete Query: $\pi_{g.guide_name}(\sigma_{student_count=S2.max_guide_count}(G2) (\gamma_{g.guide_name, count(*) \rightarrow student_count}(\sigma_{s.dept='CSE'}(guide \bowtie_{g.student_phone=s.phone} student \ s))))$

10. Sql Query:

with ranks as (select s.name, s.dept, s.gpa, row_number() over (partition by s.dept order by s.gpa) as rn, count(*) over (partition by s.dept) as total_count from student s), median_gpa as (select dept, case when total_count % 2 = 1 then max(case when rn = (total_count + 1) / 2 then gpa end) else avg(case when rn in (total_count / 2, total_count / 2 + 1) then gpa end) end as median from ranks group by dept, total_count) select s.name, s.dept from student s join median_gpa mg on s.dept = mg.dept where s.gpa < mg.median

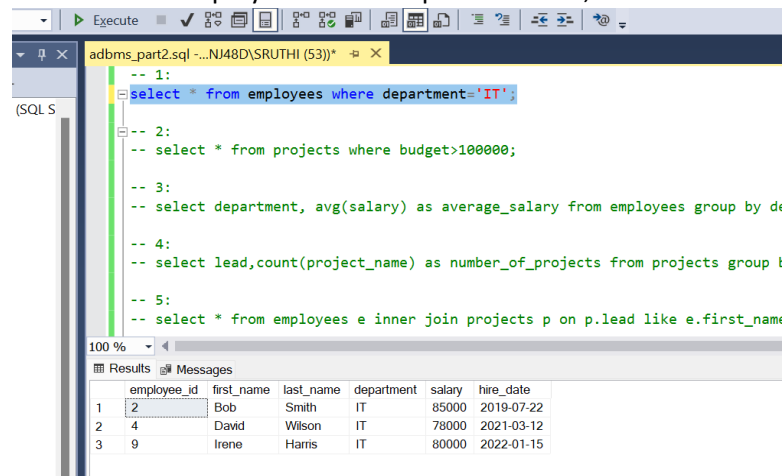
Relational Algebra:

$S1 \leftarrow \gamma_{s.name, s.dept, s.gpa, ROW_NUMBER() \rightarrow rn, COUNT(*) \rightarrow total_count}(\sigma(s)(student))$

$\pi_{s.name, s.dept}(\sigma_{s.gpa < mg.median}(students \bowtie_{s.dept=mg.dept} (\gamma_{dept, CASE WHEN total_count \bmod 2 = 1 THEN MAX(gpa) WHEN ELSE AVG(gpa)(S1))))$

Part2:

1. select * from employees where department='IT';



The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```
-- 1:
select * from employees where department='IT';

-- 2:
select * from projects where budget>100000;

-- 3:
select department, avg(salary) as average_salary from employees group by department;

-- 4:
select lead, count(project_name) as number_of_projects from projects group by lead;

-- 5:
select * from employees e inner join projects p on p.lead like e.first_name;
```

The results pane shows the output of the first query, displaying a table with 5 columns: employee_id, first_name, last_name, department, salary, and hire_date. The table contains 3 rows of data:

employee_id	first_name	last_name	department	salary	hire_date
2	Bob	Smith	IT	85000	2019-07-22
4	David	Wilson	IT	78000	2021-03-12
9	Irene	Harris	IT	80000	2022-01-15

2. select * from projects where budget>100000;

```
adbms_part2.sql -...NJ48D\SRUTHI (53))* -> X
-- 1:
-- select * from employees where department='IT';

-- 2:
select * from projects where budget>100000;

-- 3:
-- select department, avg(salary) as average_salary from employees;

-- 4:
-- select lead,count(project_name) as number_of_projects from pro;

-- 5:
-- select * from employees e inner join projects p on p.lead like
```

100 %

Results Messages

	project_id	project_name	lead	budget
1	101	Alpha	Alice Johnson	150000
2	102	Beta	Bob Smith	200000
3	103	Gamma	Carol Davis	120000

- select department, avg(salary) as average_salary from employees group by department;

```
adbms_part2.sql -...NJ48D\SRUTHI (53))* -> X
-- 1:
-- select * from employees where department='IT';

-- 2:
-- select * from projects where budget>100000;

-- 3:
select department, avg(salary) as average_salary from employees group by department;

-- 4:
-- select lead,count(project_name) as number_of_projects from projects group by lead h

-- 5:
-- select * from employees e inner join projects p on p.lead like e.first_name+' '+e.l
```

100 %

Results Messages

	department	average_salary
1	Finance	96500
2	HR	72333
3	IT	81000
4	Marketing	69500

- select lead,count(project_name) as number_of_projects from projects group by lead having count(project_name)>1;

```
adbms_part2.sql -...NJ48D\SRUTHI (53))* -> X
-- 1:
-- select * from employees where department='IT';

-- 2:
-- select * from projects where budget>100000;

-- 3:
-- select department, avg(salary) as average_salary from employees group by department;

-- 4:
select lead,count(project_name) as number_of_projects from projects group by lead having count(project_name)>1;

-- 5:
-- select * from employees e inner join projects p on p.lead like e.first_name+' '+e.last_name where p.budget>50000
```

100 %

Results Messages

	lead	number_of_projects
--	------	--------------------

- select * from employees e inner join projects p on p.lead=e.first_name+' '+e.last_name where p.budget>50000 and e.hire_date >= DATEADD(YEAR, -5, GETDATE())

adbms_part2.sql -...NJ48D\SRUTHI (53))*

```
-- 4:
-- select lead,count(project_name) as number_of_projects from projects group by lead havi

-- 5:
select * from employees e inner join projects p on p.lead=e.first_name+' '+e.last_name w

-- 6:
-- insert into employees values(11,'Ron','Doe','Finance',90000,'2023-03-12'),
-- (12,'Jane','Smith','IT',85000,'2020-08-01'),
-- (13,'Michael','Johnson','HR',75000,'2019-07-23')
-- insert into projects values(106,'Zeta','Frank Taylor',85000),
-- (107,'Eta','Grace Lee',100000),
-- (108,'Theta','Henry Martin',90000),
-- (109,'Iota','Irene Harris',150000)
```

100 %

Results Messages

	employee_id	first_name	last_name	department	salary	hire_date	project_id	project_name	lead	budget
1	1	Alice	Johnson	HR	70000	2020-01-15	101	Alpha	Alice Johnson	150000
2	4	David	Wilson	IT	78000	2021-03-12	104	Delta	David Wilson	80000

6. insert into employees values(11,'Ron','Doe','Finance',90000,'2023-03-12'), (12,'Jane','Smith','IT',85000,'2020-08-01'), (13,'Michael','Johnson','HR',75000,'2019-07-23')

adbms_part2.sql -...NJ48D\SRUTHI (53))*

```
-- 5:
-- select * from employees e inner join projects p on p.lead=e.first_name+' '+e.last_name w

-- 6:
insert into employees values(11,'Ron','Doe','Finance',90000,'2023-03-12'),
(12,'Jane','Smith','IT',85000,'2020-08-01'),
(13,'Michael','Johnson','HR',75000,'2019-07-23')
-- insert into projects values(106,'Zeta','Frank Taylor',85000),
-- (107,'Eta','Grace Lee',100000),
-- (108,'Theta','Henry Martin',90000),
-- (109,'Iota','Irene Harris',150000)
```

100 %

Messages

(3 rows affected)

Completion time: 2024-09-24T15:39:10.4483227-05:00

insert into projects values(106,'Zeta','Frank Taylor',85000),(107,'Eta','Grace Lee',100000), (108,'Theta','Henry Martin',90000), (109,'Iota','Irene Harris',150000)

```
-- 6:
-- insert into employees values(11,'Ron','Doe','Finance',90000,'2023-03-12'),
-- (12,'Jane','Smith','IT',85000,'2020-08-01'),
-- (13,'Michael','Johnson','HR',75000,'2019-07-23')
insert into projects values(106,'Zeta','Frank Taylor',85000),
(107,'Eta','Grace Lee',100000),
(108,'Theta','Henry Martin',90000),
(109,'Iota','Irene Harris',150000)
```

%

Messages

(4 rows affected)

Completion time: 2024-09-24T15:42:15.1294585-05:00

select * from employees

```
-- (107, 'Eta', 'Grace Lee', 100000),
-- (108, 'Theta', 'Henry Martin', 90000),
-- (109, 'Iota', 'Irene Harris', 150000)
select * from employees
```

	employee_id	first_name	last_name	department	salary	hire_date
1	1	Alice	Johnson	HR	70000	2020-01-15
2	2	Bob	Smith	IT	85000	2019-07-22
3	3	Carol	Davis	Finance	95000	2018-07-22
4	4	David	Wilson	IT	78000	2021-03-12
5	5	Eva	Brown	HR	72000	2022-06-20
6	6	Frank	Taylor	Marketing	68000	2023-05-01
7	7	Grace	Lee	Finance	98000	2017-10-25
8	8	Henry	Martin	Marketing	71000	2020-12-10
9	9	Irene	Harris	IT	80000	2022-01-15
10	10	John	Clark	HR	75000	2019-08-30
11	11	Ron	Doe	Finance	90000	2023-03-12
12	12	Jane	Smith	IT	85000	2020-08-01
13	13	Michael	Johnson	HR	75000	2019-07-23

select * from projects

```
-- (109, 'Iota', 'Irene Harris', 150000)
-- select * from employees
select * from projects
```

	project_id	project_name	lead	budget
1	101	Alpha	Alice Johnson	150000
2	102	Beta	Bob Smith	200000
3	103	Gamma	Carol Davis	120000
4	104	Delta	David Wilson	80000
5	105	Epsilon	Eva Brown	50000
6	106	Zeta	Frank Taylor	85000
7	107	Eta	Grace Lee	100000
8	108	Theta	Henry Martin	90000
9	109	Iota	Irene Harris	150000

7. update projects set budget=160000 where project_name='Alpha'

```
-- 7:
update projects set budget=160000 where project_name='Alpha'
```

(1 row affected)

Completion time: 2024-09-24T15:48:37.3774906-05:00

```
-- select * from projects
-- 7:
-- update projects set budget=160000 where project_name='Alpha'
select * from projects
```

	project_id	project_name	lead	budget
1	101	Alpha	Alice Johnson	160000
2	102	Beta	Bob Smith	200000
3	103	Gamma	Carol Davis	120000
4	104	Delta	David Wilson	80000
5	105	Epsilon	Eva Brown	50000
6	106	Zeta	Frank Taylor	85000
7	107	Eta	Grace Lee	100000
8	108	Theta	Henry Martin	90000
9	109	Iota	Irene Harris	150000

8. delete from projects where last_budget_increase>dateadd(year,-3,getdate())

```
-- 7:
-- update projects set budget=160000 where project_name='Alpha'
-- select * from projects

-- 8:
delete from projects where last_budget_increase>dateadd(year,-3,getdate())
-- select * from projects

-- 9:
-- select * from employees where department='Marketing' and salary>(select
-- 10:
-- select e.* from employees e inner join projects p on e.first_name+' '+e.

-- 8:
-- delete from projects where last_budget_increase>dateadd(year,-3,getdate())
-- select * from projects

-- 9:
-- select * from employees where department='Marketing' and salary>(select AVG
-- 10:
-- select e.* from employees e inner join projects p on e.first_name+' '+e.

100 %
Results Messages
project_id project_name lead budget last_budget_increase
1 101 Alpha Alice Johnson 160000 2020-01-15
2 102 Beta Bob Smith 200000 2019-07-22
3 104 Delta David Wilson 80000 2018-03-12
4 106 Zeta Frank Taylor 85000 2020-01-15
5 107 Eta Grace Lee 100000 2019-07-22
6 109 Iota Irene Harris 150000 2018-03-12
```

9. select * from employees where department='Marketing' and salary>(select AVG(salary) from employees where department='Marketing')

```
-- 9:
select * from employees where department='Marketing' and salary>(select AVG(salary) from employees where department='Marketing')

-- 10:
-- select e.* from employees e inner join projects p on e.first_name+' '+e.last_name= p.lead where datediff(day,e.hire_date,getdate())>1095;

-- extra:
-- select * from employee_skills
--with employee_skillsets as(select employee id,string agg(Skill, ' ') within group(order by Skill) as Skills,count(distinct Skill) as SkillCount from Employee_Skills group by employee_id having count(distinct Skill)>=3) select employee_id,skills from employee_skillsets where Skills in(select Skills from employee_skillsets group by Skills having count(*)=1)

10 %
Results Messages
employee_id first_name last_name department salary hire_date
8 Henry Martin Marketing 71000 2020-12-10
```

10. select e.* from employees e inner join projects p on e.first_name+' '+e.last_name= p.lead where datediff(day,e.hire_date,getdate())>1095;

```
-- 10:
select e.* from employees e inner join projects p on e.first_name+' '+e.last_name= p.lead where datediff(day,e.hire_date,getdate())>1095;

-- extra:
-- select * from employee_skills
--with employee_skillsets as(select employee id,string agg(Skill, ' ') within group(order by Skill) as Skills,count(distinct Skill) as SkillCount from Employee_Skills group by employee_id having count(distinct Skill)>=3) select employee_id,skills from employee_skillsets where Skills in(select Skills from employee_skillsets group by Skills having count(*)=1)

0 %
Results Messages
employee_id first_name last_name department salary hire_date
1 Alice Johnson HR 70000 2020-01-15
2 Bob Smith IT 85000 2019-07-22
4 David Wilson IT 78000 2021-03-12
7 Grace Lee Finance 98000 2017-10-25
```

Extra Credit:

with employee_skillsets as(select employee_id,string_agg(Skill, ' ') within group(order by Skill) as Skills,count(distinct Skill) as SkillCount from Employee_Skills group by employee_id having count(distinct Skill)>=3) select employee_id,skills from employee_skillsets where Skills in(select Skills from employee_skillsets group by Skills having count(*)=1)

```
-- extra:
select * from employee_skills
--with employee_skillsets as(select employee id,string agg(Skill, ' ') within group(order by Skill) as Skills,count(distinct Skill) as SkillCount from Employee_Skills group by employee_id having count(distinct Skill)>=3) select employee_id,skills from employee_skillsets where Skills in(select Skills from employee_skillsets group by Skills having count(*)=1)

100 %
Results Messages
employee_id skill
1 1 SQL
2 1 Python
3 1 Data Analysis
4 2 Java
5 2 Python
6 3 SQL
7 3 Machine Learning
8 4 Python
9 4 SQL
10 5 Data Analysis
11 5 Project Management

-- extra:
-- select * from employee_skills
with employee_skillsets as(select employee_id,string_agg(Skill, ' ') within group(order by Skill) as Skills,count(distinct Skill) as SkillCount from Employee_Skills group by employee_id having count(distinct Skill)>=3) select employee_id,skills from employee_skillsets where Skills in(select Skills from employee_skillsets group by Skills having count(*)=1)

00 %
Results Messages
employee_id skills
1 1 Data Analysis, Python, SQL
```