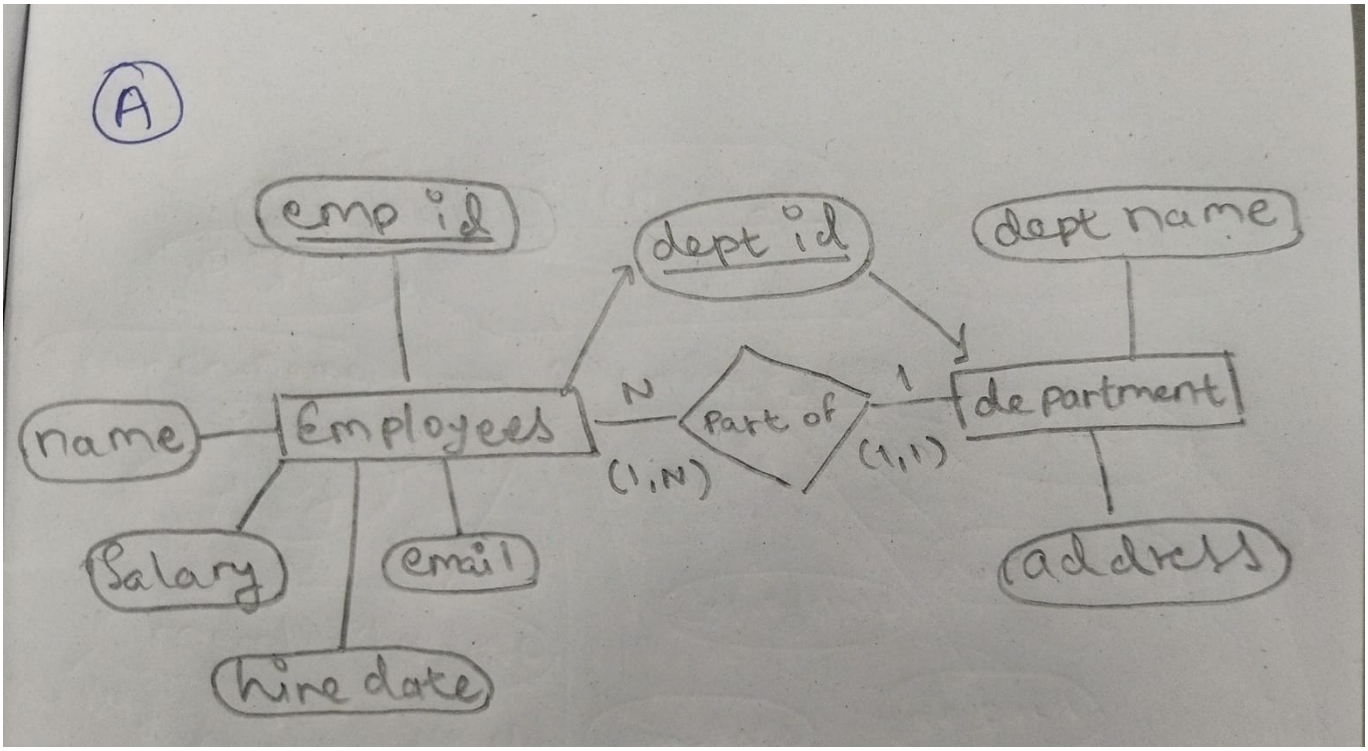
DBMS LAB 5

Siddharth Karmokar

- 123CS0061

Employee-Department Management

Part A: ER Diagram



- Part B

```
create table departments(
    dept_id number primary key,
    dept_name  varchar2(100),
    address varchar2(100)
);

select * from departments;
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.013 seconds

DEPT_ID	DEPT_NAME	ADDRESS

```
alter table employees
modify email varchar2(60);
```

- Part C

```
insert into departments values(1, 'Computer Science', 'Kurnool');
insert into departments values(2, 'Arificial Intelligence', 'Kurnool');
insert into departments values(3, 'Mechanical', 'Kurnool');

select * from employees;
select * from departments;
```

Query Result 2 — All Rows Fetched: 3 in 0.007 seconds

	DEPT_ID	DEPT_NAME	ADDRESS
1	1	Computer Science	Kurnool
2	2	Arificial Intelligence	Kurnool
3	3	Mechanical	Kurnool

All Rows Fetched: 8 in 0.003 seconds

	EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID
1	101	alice johnson	55000	15-06-22	alice.johnson@example.com	1
2	102	bob smith	60000	20-03-21	bob.smith@example.com	2
3	103	carol davis	58000	10-01-23	carol.davis@example.com	3
4	104	david lee	62000	05-11-20	david.lee@example.com	1
5	105	emma watson	54000	01-09-22	emma.watson@example.com	2
6	106	frank martin	61000	18-12-21	frank.martin@example.com	3
7	107	grace hall	57000	25-07-23	grace.hall@example.com	1
8	108	harry potter	63000	12-05-20	harry.potter@example.com	2

```
update employees
set salary = salary + (0.1)*salary
where dept_id = 1;

select * from employees;
select * from departments;
```

Query Result 2 — All Rows Fetched: 8 in 0.002 seconds

	EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID
1	101	alice johnson	60500	15-06-22	alice.johnson@example.com	1
2	102	bob smith	60000	20-03-21	bob.smith@example.com	2
3	103	carol davis	58000	10-01-23	carol.davis@example.com	3
4	104	david lee	68200	05-11-20	david.lee@example.com	1
5	105	emma watson	54000	01-09-22	emma.watson@example.com	2
6	106	frank martin	61000	18-12-21	frank.martin@example.com	3
7	107	grace hall	62700	25-07-23	grace.hall@example.com	1
8	108	harry potter	63000	12-05-20	harry.potter@example.com	2

```
delete from employees
where salary <= 25000;

select * from employees;
```

Task completed in 0.027 seconds

```
create table stg_employees(
    emp_id number primary key,
    name varchar2(100),
    salary number,
    hire_date date,
    email varchar2(100),
    dept_id number not null,
    foreign key(dept_id) references departments(dept_id)
);
```

Query Result 3 — All Rows Fetched: 2 in 0.003 seconds

	EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID
1	108	harry potter	63000	12-05-20	harry.potter@example.com	1
2	110	giny potter	69000	12-05-20	giny.potter@example.com	2

```
merge into employees tar
using stg_employees src
on (tar.emp_id = src.emp_id)
when matched then
    update set tar.dept_id = src.dept_id
when not matched then
    insert (emp_id, name, salary, hire_date, email, dept_id)
    values (src.emp_id, src.name, src.salary, src.hire_date, src.email, src.dept_id);
```

Query Result 3 — All Rows Fetched: 9 in 0.001 seconds

	EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID
1	101	alice johnson	60500	15-06-22	alice.johnson@example.com	1
2	102	bob smith	60000	20-03-21	bob.smith@example.com	2
3	103	carol davis	58000	10-01-23	carol.davis@example.com	3
4	104	david lee	68200	05-11-20	david.lee@example.com	1
5	105	emma watson	54000	01-09-22	emma.watson@example.com	2
6	106	frank martin	61000	18-12-21	frank.martin@example.com	3
7	107	grace hall	62700	25-07-23	grace.hall@example.com	1
8	108	harry potter	63000	12-05-20	harry.potter@example.com	1
9	110	giny potter	69000	12-05-20	giny.potter@example.com	2

- Part D

```
select * from employees
order by salary desc;

select * from stg_employees;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 9 in 0.002 seconds

	EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID
1	110	giny potter	69000	12-05-20	giny.potter@example.com	2
2	104	david lee	68200	05-11-20	david.lee@example.com	1
3	108	harry potter	63000	12-05-20	harry.potter@example.com	1
4	107	grace hall	62700	25-07-23	grace.hall@example.com	1
5	106	frank martin	61000	18-12-21	frank.martin@example.com	3
6	101	alice johnson	60500	15-06-22	alice.johnson@example.com	1
7	102	bob smith	60000	20-03-21	bob.smith@example.com	2
8	103	carol davis	58000	10-01-23	carol.davis@example.com	3
9	105	emma watson	54000	01-09-22	emma.watson@example.com	2

```
select distinct name from employees;

select * from stg_employees;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 9 in 0.002 seconds

	NAME
1	alice johnson
2	bob smith
3	carol davis
4	david lee
5	emma watson
6	frank martin
7	grace hall
8	harry potter
9	giny potter

```
select * from employees
where salary > 30000 and salary < 50000;

select * from stg_employees;
select * from employees;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 0 in 0.002 seconds

EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID

```
select * from employees
where lower(name) like 'a%';

select * from stg_employees;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 1 in 0.003 seconds

	EMP_ID	NAME	SALARY	HIRE_DATE	EMAIL	DEPT_ID
1	101	alice johnson	60500	15-06-22	alice.johnson@example.com	1

```
select dept_id, count(*) as num_emp from employees
group by dept_id;

select * from stg_employees;
select * from employees;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 3 in 0.002 seconds

	DEPT_ID	NUM_EMP
1	1	4
2	2	3
3	3	2

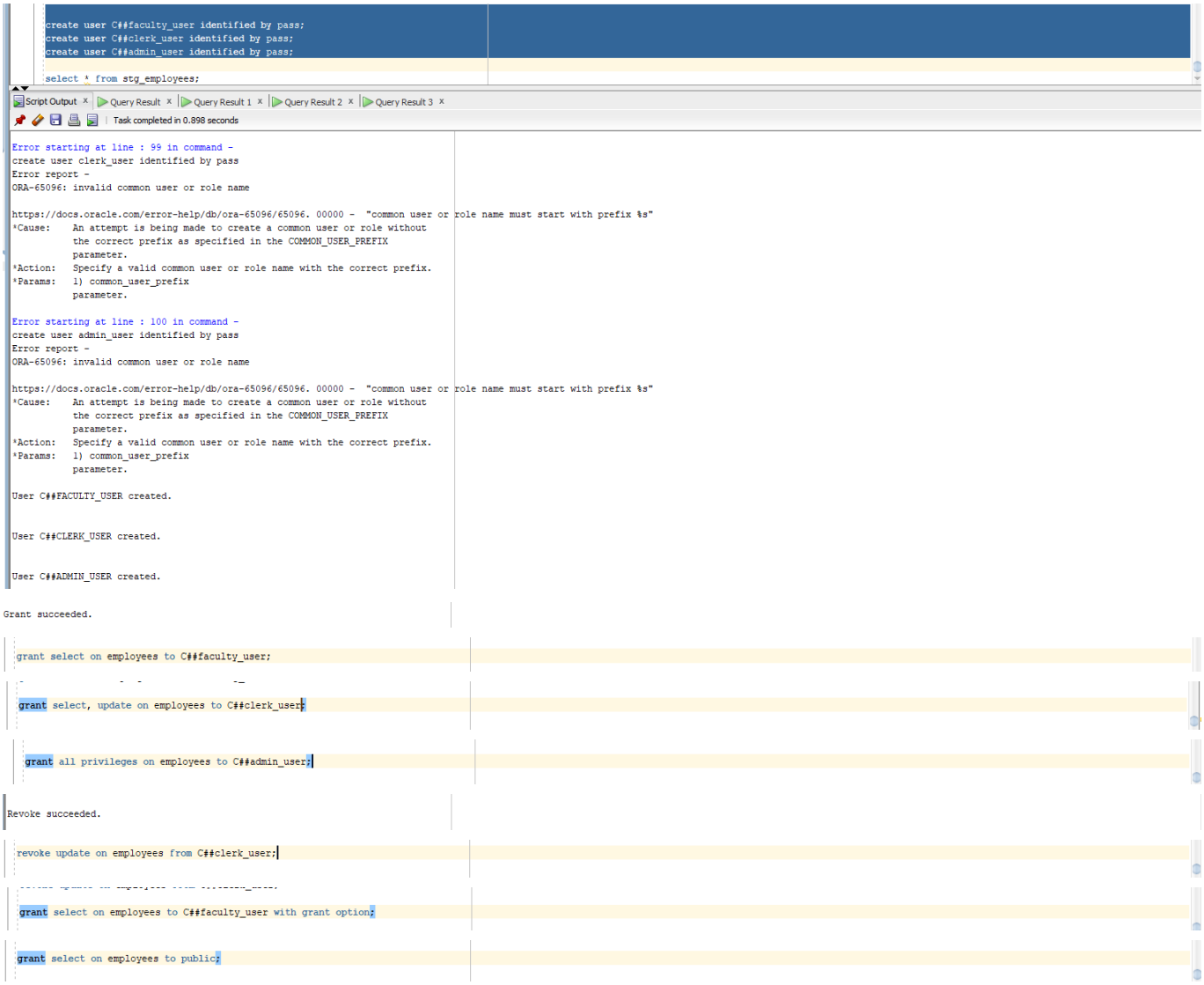
```
select dept_id, avg(salary) as avg_salary from employees
group by dept_id
having avg(salary) > 40000;

select * from stg_employees;
select * from employees;
select * from departments;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 3 in 0.002 seconds

	DEPT_ID	AVG_SALARY
1	1	63600
2	2	61000
3	3	59500

- Part E



- Part F

22. What happens if `faculty_user` tries to insert into `EMPLOYEES`?

> If `faculty_user` **does not have the INSERT privilege** on the `EMPLOYEES` table, the operation will fail with a **permission error**:

```
ORA-01031: insufficient privileges
```

> To allow insertion, the following must be granted:

```
grant insert on employees to C##faculty_user;
```

---

23. Can `clerk_user` still update after you revoked the privilege?

> **No**. Once the `UPDATE` privilege is revoked, `clerk_user` **can no longer perform update operations** on the table. Any attempt will result in a **permission denied error**:

```
ORA-01031: insufficient privileges
```

---

24. Why is `WITH GRANT OPTION` considered dangerous in real systems?

> Because it allows a user to **grant the same privilege to others**, it can:

- Bypass controlled access,
- Lead to **unauthorized privilege escalation**,
- Make tracking and revoking permissions harder,
- Pose **security risks** in multi-user environments.

---

25. What is the effect of granting privileges to `PUBLIC`?

> Granting to `PUBLIC` makes the privilege available to **all users in the database** — present and future.
> It's **high-risk** because:

- Even unauthorized users gain access,
- It breaks the principle of least privilege,
- It may expose sensitive data or allow unintended actions.