Chandigarh College of Engineering & Technology (Degree Wing)

Department of Computer Science and Engineering

Database Systems (Practical)

CS 352

Practical - 8

DOP: - 20/09/2024 **DOS:-** / /2024

Submitted By:

Abhay Pratap Singh
(Roll No: - CO23306)

Submitted To:

Dr. Dheerendra Singh
CSE Department

AIM :- Implementation of at least five different SQL nested sub queries in select clause on each Banking-University and DB Project assigned.

STEP 1 :- Implementation of at least five different SQL nested sub queries on banking data

Query 1 :-

```
SELECT
   account_number,
   branch_name,
   balance,
   (SELECT COUNT(*)
   FROM depositor
   WHERE depositor.account_number = account.account_number) AS depositor_count
FROM
   account;
```

This query retrieves the account number and the count of depositors for each account.

account_number	branch_name	balance	depositor_count
A-101	Downtown	500	1
A-102	Perryridge	400	1
A-201	Brighton	900	1
A-215	Mianus	700	1
A-217	Brighton	750	1
A-222	Redwood	700	1
A-305	Round Hill	350	1

Query 2 :-

```
SELECT
   customer_name,
   loan_number,
   (SELECT amount
    FROM loan
    WHERE loan.loan_number = borrower.loan_number) AS loan_amount
FROM
   borrower;
```

This query retrieves each borrower's name and the amount of their loan.

customer_name	loan_number	loan_amount
Smith	L-11	900
Jackson	L-14	1500
Hayes	L-15	1500
Adams	L-16	1300
Williams	L-17	1000
Smith	L-23	2000
Curry	L-93	500

Query 3 :-

```
SELECT
   branch_name,
   (SELECT SUM(amount)
    FROM loan
    WHERE loan.branch_name = branch.branch_name) AS total_loans
FROM
   branch;
```

This query retrieves each branch's name and the total amount of loans issued from that branch.

branch_name	total_loans
Brighton	*NULL*
Downtown	2500
Mianus	500
North Town	*NULL*
Perryridge	2800
Pownal	*NULL*
Redwood	2000
Round Hill	900

Query 4 :-

```
SELECT
   branch_name,
   (SELECT COUNT(*)
    FROM account
    WHERE account.branch_name = branch.branch_name) AS total_accounts
FROM
   branch;
```

This query retrieves each branch's name along with the total number of accounts associated with that branch

branch_name	total_accounts
Brighton	2
Downtown	1
Mianus	1
North Town	0
Perryridge	1
Pownal	0
Redwood	1
Round Hill	1

Query 5 :-

```
SELECT
   branch_name,
   (SELECT AVG(balance)
    FROM account
    WHERE account.branch_name = branch.branch_name) AS average_balance
FROM
   branch;
```

This query retrieves each branch's name along with the average balance of accounts associated with that branch.

branch_name	average_balance
Brighton	825.0000
Downtown	500.0000
Mianus	700.0000
North Town	*NULL*
Perryridge	400.0000
Pownal	*NULL*
Redwood	700.0000
Round Hill	350.0000

STEP 2 :- Implementation of at least five different SQL nested sub queries on University data

Query 1 :-

```
SELECT
   dept_name,
   (SELECT COUNT(*)
    FROM course
    WHERE course.dept_name = department.dept_name) AS course_count
FROM
   department;
```

This query retrieves each department's name and the count of courses offered in that department.

dept_name	course_count
Biology	3
Comp. Sci.	5
Elec. Eng.	1
Finance	1
History	1
Music	1
Physics	1

Query 2:-

```
SELECT
   name,
   salary,
   (SELECT budget
    FROM department
    WHERE department.dept_name = instructor.dept_name) AS dept_budget
FROM
   instructor;
```

This query retrieves each instructor's name, salary, and the budget of their department.

name	salary	dept_budget
Dr Dheerendra Singh	154239.96	*NULL*
vicky	163909.05	*NULL*
Srinivasan	75245.63	*NULL*
taran	163909.05	80000.00
Wu	104186.25	120000.00
Mozart	46305.00	80000.00
Mozart	46305.00	80000.00
Wu	138910.50	*NULL*
Einstein	107879.63	70000.00
El Said	69457.50	50000.00
Gold	100713.38	70000.00
Katz	86821.88	*NULL*
Califieri	71772.75	50000.00
Singh	92610.00	120000.00
Crick	83349.00	90000.00
Brandt	104472.90	*NULL*
Kim	92610.00	85000.00

Query 3:-

```
SELECT
   building,
   (SELECT SUM(capacity)
    FROM classroom c
    WHERE c.building = classroom.building) AS total_capacity
FROM
   classroom
GROUP BY
   building;
```

This query retrieves each building's name and the total capacity of classrooms within that building.

building	total_capacity
Packard	500
Painter	10
Taylor	70
Watson	80

Query 4 :-

```
SELECT
   dept_name,
   (SELECT AVG(credits)
    FROM course
    WHERE course.dept_name = department.dept_name) AS average_credits
FROM
   department;
```

This query retrieves each department's name and the average credits for courses offered in that department.

dept_name	average_credits
Biology	3.6667
Comp. Sci.	3.4000
Elec. Eng.	3.0000
Finance	3.0000
History	3.0000
Music	3.0000
Physics	4.0000

Query 5 :-

```
SELECT
   dept_name,
   (SELECT AVG(salary)
    FROM instructor
    WHERE instructor.dept_name = department.dept_name) AS average_salary
FROM
   department;
```

This query retrieves each department's name and the average salary of instructors in that department.

dept_name	average_salary
Biology	83349.000000
Comp. Sci.	*NULL*
Elec. Eng.	92610.000000
Finance	98398.125000
History	70615.125000
Music	85506.350000
Physics	104296.505000

STEP 3 :- Implementation of at least five different SQL nested sub queries on Computational data of all Staff (Project no :- 7)

Schema :-

- details_teaching_official(NAME, DESIGNATION, DEPARTMENT, DESIGNATION-ID, DEPARTMENT-ID, ROOM_NO., JOINING_DATE, EXPERIENCE_MONTHS, ASSOCIATION_TYPE)
- detail_teaching_personal (NAME, AGE, GENDER, PHONE_NUMBER, QUALIFICATION,E-MAIL)
- details_non_teaching (NAME, GENDER, EMAIL_ID, DESIGNATION-ID, DEPARTMENT-ID)

Query 1:-

```
SELECT
   DEPARTMENT,
   (SELECT COUNT(*)
    FROM details_teaching_official t
    WHERE t.DEPARTMENT = details_teaching_official.DEPARTMENT) AS staff_count
FROM
   details_teaching_official
GROUP BY
   DEPARTMENT;
```

Count of Teaching Staff in Each Department

DEPARTMENT	staff_count
Applied Science	4
Civil Engg.	7
Computer Science Engg.	10
Electronics and Communications Engg.	10
Mechanical Engg.	8

Query 2:-

```
SELECT
   t.DEPARTMENTID,
   (SELECT SUM(EXPERIENCE_MONTHS)
    FROM details_teaching_official
    WHERE DEPARTMENTID = t.DEPARTMENTID) AS TOTAL_EXPERIENCE
FROM
   details_teaching_official t
GROUP BY
   t.DEPARTMENTID;
```

returns a list of unique department IDs along with the total experience

DEPARTMENTID	TOTAL_EXPERIENCE
1	2485
2	2175
3	1795
4	1604
5	430

Query 3:-

```
SELECT
   t.DEPARTMENTID,
   t.NAME,
   p.AGE
FROM
   details_teaching_official t
JOIN
   details_teaching_personal p ON t.NAME = p.NAME
WHERE
   (t.DEPARTMENTID, p.AGE) IN (
```

```
    SELECT
      DEPARTMENTID,
      MAX(AGE)
    FROM
      details_teaching_personal p2
    JOIN
      details_teaching_official t2 ON p2.NAME = t2.NAME
    GROUP BY
      t2.DEPARTMENTID
);
```

Returns person with max age in each department

DEPARTMENTID	NAME	AGE
4	Aradhana Mehta	57
2	Davinder Singh Saini	46
3	Jatinder Madan	52
5	Parul Aggarwal	43
1	Ram Bahadur Patel	56

Query 4:-

```
SELECT
   p.QUALIFICATION,
   (SELECT COUNT(*)
    FROM details_teaching_personal p2
    JOIN details_teaching_official t2 ON p2.NAME = t2.NAME
    WHERE p2.QUALIFICATION = p.QUALIFICATION) AS NUMBER_OF_PERSONS
FROM
   details_teaching_personal p
GROUP BY
   p.QUALIFICATION;
```

Returns qualification and no of persons that holds it

QUALIFICATION	NUMBER_OF_PERSONS
M.E.	2
M.Sc.	1
M.Tech	9
Ph.D	27

Query 5:-

```
SELECT
   GENDER,
   COUNT(*) AS NUMBER_OF_PERSONS
FROM
   (SELECT
      GENDER
    FROM
      details_non_teaching_staff) AS subquery
GROUP BY
   GENDER;
```

GENDER	NUMBER_OF_PERSONS
F	10
M	22