SE302T : Database Management Systems

8.7
10

Quiz # 3

Name: Zunaira Abdul Aziz **Roll #:** BSSE23058-A.

Time: 25 minutes **Marks:** 20

Table: EMPLOYEE

EmpID	FirstName	LastName	DeptID	Salary	HireDate	ManagerID
101	Zara	Khan	10	5000	2020-02-15	NULL
102	Ali	Raza	20	3400	2019-11-01	101
103	Hira	Sheikh	30	2800	2021-03-12	102
104	Usman	Tariq	NULL	2000	2022-05-20	101
105	Saba	Rehman	20	1200	2023-01-10	102

Join

Table: DEPARTMENT

DeptID	DeptName	Location	Budget
10	HR	Lahore	150000
20	IT	Islamabad	300000
30	Marketing	Karachi	200000
40	Operations	Peshawar	100000

Table: SALARY_GRADE

Grade	MinSalary	MaxSalary
A	0	1999
B	2000	2999
C	3000	3999
D	4000	5999

Question # 1: Write a SQL query to retrieve the full name (FirstName + LastName), department name, and location of all employees **who joined before 2022** and work in departments located in **Islamabad or Karachi**. Use **aliases** for tables and qualify ambiguous column.**(Marks-3)** (CLO-3)

SELECT ~~e.~~ SUM(e.FirstName + e.LastName). AS fullName , d.DeptName
d.location
FROM. EMPLOYEE e.
INNER JOIN. Department d ON. e.DeptID = d.DeptID.
WHERE e.HireDate ~~HireDate (~~ < (YEAR.('2022') AND d. ~~DeptName~~ Location. IN ('ISLAMABAD'
'KARACHI')
GROUP BY. e.HireDate , d.Location , d. deptName.

?

1.1

2

Question # 2: (Marks-~~4~~) (CLO-3)

(a) Write a query to display all departments and the names of employees working in them, **including departments with no employees.**

SELECT ~~d.~~ d.deptID, d.DeptName, e.firstName, e.LastName, e.employeeID

FROM ~~employee e,~~ Department d.

LEFT ~~INNER INSERT~~ JOIN. EMPLOYEE e.

ON d.DeptID = e.DeptID.

WHERE e.DeptID IS NULL.

0.9

(b) Modify the query to also include the **budget** of the department and display “No Employee” if no one is assigned.

~~SELECT~~ SELECT d.budget.

FROM department d

INNER JOIN ~~Department~~ SALARY_GRADE. sg

d. Budget = Sg. Budget.

?

?

0.2

Question # 3: Write a query using a **non-equijoin** to list each employee’s full name, salary, and the corresponding **salary grade** (A–D). Sort the results by salary in descending order. **(~~3~~ Marks)** (CLO-3)

1.5

SELECT sum(e.firstname + e.LastName) AS fullname, e. salary, sg. ~~salary~~ grade

FROM. employee e. ~~and~~.

~~JOIN~~. , salary grade sg

WHERE e. salary BETWEEN sg. Minsalary AND sg. Maxsalary.

GROUP BY e.salary. , sg. grade

ORDER.BY e.salary. DES.

1.3

Question # 4: You are working with the following tables: **(~~6~~ Marks)** 3 (CLO-3)

- EMPLOYEE (5 rows)
- DEPARTMENT (4 rows)

Accidentally, a junior developer runs the following SQL query:

```
SELECT e.FirstName, d.DeptName, e.Salary
FROM EMPLOYEE e, DEPARTMENT d;
```

(a) Explain in detail:

- What type of join this is implicitly creating → INNER JOIN, as it is created by default.
- How many rows will be returned and why?

↳ - 5 rows, as it will ~~dis~~ JOIN the DeptID and there are five employees and no condition is given. so will display all the employees.

0.5

(b) Now assume that the developer adds a WHERE clause to filter salaries above 3000:

```
SELECT e.FirstName, d.DeptName, e.Salary
FROM EMPLOYEE e, DEPARTMENT d
WHERE e.Salary > 3000;
```

- How many rows will be returned now? 2 rows.
- Has the Cartesian product problem been fixed? Why or why not? ~~Yes, as it doesnt Atturn all the 5×4 = 20 rows.~~

Yes the cartesian product problem solves as not all the rows (5×4 = 20 rows) are displayed due to the condition of where clause being amplied.

2

(c) Write the correct INNER JOIN version of the query that:

- Lists employee full name, department name, location
- Only includes employees who are **assigned to a department**
- Orders results by salary descending.

```
select  sum(e.firstName + e.LastName) AS full name,
        d.deptName, d.Location.
FROM    Employee e, Department d.
INNER JOIN . Department.d. ON e.deptID = d.deptID.
WHERE   d e.deptID is NOT NULL
```

0.9

Question # 5: (4 Marks) **(CLO-3)**

(a) Write a query using a **self-join** to list each employee and their manager's full name.

```
SELECT   a.empID AS NORMALEMPLOYEES, b.EMPID AS
           MANAGER ID, SUM(e.FirstName + e.LastName) AS
         FULL_name..

   FROM    Employee e.

INNER. JOIN    EMPLOYEE e.
   ON.       e.empID = e.employ ID
```

0.4

(b) Modify the query to only show employees hired **after** their managers.

```
WHERE  a.Hirdate.  > b.HireDate.
```

0.4