|  |  |  |
| --- | --- | --- |
| **Session Topic/Title** | **Session No.** | **Session Duration (Minutes)** |
| Complex Queries | 9 | 180 |

|  |  |
| --- | --- |
| 1. **Session Outcomes** |  |

1. Develop databases and associated queries using SQL **(SO:2; PI:2.2)**

|  |  |
| --- | --- |
| 1. **Tool(s)/Software** |  |

MySQL Workbench

|  |  |
| --- | --- |
| 1. **Procedural steps (Tasks)** |  |

**Description**

**Exercise (1):**

1. **Retrieve the names of employees in department 2 who work more than 5 hours per week on the 'Constructions' project.**

SELECT LNAME, FNAME

FROM EMPLOYEE, WORKS\_ON, PROJECT

WHERE DNO=2 AND SSN=ESSN AND PNO=PNUMBER AND PNAME='Constructions' AND HOURS>5;

Another possible SQL query uses nesting as follows:

SELECT LNAME, FNAME

FROM EMPLOYEE

WHERE DNO=2 AND SSN IN ( SELECT ESSN

FROM WORKS\_ON

WHERE HOURS>5 AND PNO IN ( SELECT PNUMBER

FROM PROJECT

WHERE PNAME='Constructions' ) );

1. **List the names of employees who have a dependent with the same first name as themselves.**

SELECT LNAME, FNAME

FROM EMPLOYEE, DEPENDENT

WHERE SSN=ESSN AND FNAME=DEPENDENT\_NAME;

Another possible SQL query uses nesting as follows:

SELECT LNAME, FNAME

FROM EMPLOYEE

WHERE EXISTS ( SELECT \*

FROM DEPENDENT

WHERE FNAME=DEPENDENT\_NAME AND SSN=ESSN );

1. **Find the names of employees that are directly supervised by 'Abdulaziz Omar’.**

SELECT E.LNAME, E.FNAME

FROM EMPLOYEE E, EMPLOYEE S

WHERE S.FNAME='Abdulaziz' AND S.LNAME='Omar' AND E.SUPERSSN=S.SSN;

Another possible SQL query uses nesting as follows:

SELECT LNAME, FNAME

FROM EMPLOYEE

WHERE SUPERSSN IN ( SELECT SSN

FROM EMPLOYEE

WHERE FNAME='Abdulaziz' AND LNAME='Omar' );

1. **For each project, list the project name and the total hours per week (by all employees) spent on that project.**

SELECT PNAME, SUM (HOURS)

FROM PROJECT, WORKS\_ON

WHERE PNUMBER=PNO

GROUP BY PNAME;

1. **Retrieve the names of employees who work on every project.**

SELECT LNAME, FNAME

FROM EMPLOYEE

WHERE NOT EXISTS ( SELECT PNUMBER FROM PROJECT

WHERE NOT EXISTS ( SELECT \* FROM WORKS\_ON WHERE PNUMBER=PNO AND ESSN=SSN ) );

1. **Retrieve the names of employees who do not work on any project.**

SELECT fname,lname

from employee

where ssn not in (select essn from works\_on where ssn=essn);

1. **For each department, retrieve the department name, and the average salary of employees working in that department.**

SELECT DNAME, AVG (SALARY)

FROM DEPARTMENT, EMPLOYEE

WHERE DNUMBER=DNO

GROUP BY DNAME;

1. **Retrieve the average salary of all female employees.**

SELECT AVG (SALARY)

FROM EMPLOYEE

WHERE GENDER='F';

1. **Find the names and addresses of employees who work on at least one project located in Dammam but whose department has no location in Dammam.**

select fname, lname, address

from employee, project, department

where plocation='Dammam' and dno=dnumber and dnum = dnumber and dnum not in (select dnumber from department where dlocation='Dammam');

1. **List the last names of department managers who have no dependents**

select lname

from employee,department

where ssn=mgrssn and mgrssn not in (select essn from dependent);

**Exercise (2):**

1. **How many copies of the book titled “Science” are owned by the library branch whose name is “First branch”?**

SELECT No\_Of\_Copies

FROM ( (BOOK NATURAL JOIN BOOK\_COPIES ) NATURAL JOIN LIBRARY\_BRANCH )

WHERE Title='science' AND Branch\_Name='first branch' ;

1. **How many copies of the book titled “Technology” are owned by each library branch?**

SELECT Branch\_Name, No\_Of\_Copies

FROM ( (BOOK NATURAL JOIN BOOK\_COPIES ) NATURAL JOIN LIBRARY\_BRANCH )

WHERE Title='Technology';

1. **Retrieve the names of all borrowers who do not have any books checked out.**

SELECT Name

FROM BORROWER B

WHERE NOT EXISTs(SELECT \*

FROM BOOKs\_LOANS L

WHERE B.Card\_No = L.Card\_No);

1. **For each book that is loaned out from the "First branch" branch and whose DueDate is today, retrieve the book title, the borrower's name, and the borrower's address.**

SELECT B.Title, R.Name, R.Address

FROM BOOK B, BORROWER R, BOOKs\_LOANS BL, LIBRARY\_BRANCH LB

WHERE LB.Branch\_Name='First branch' AND LB.Branch\_Id=BL.Branch\_Id AND

BL.Due\_Date= curdate() AND BL.Card\_No=R.Card\_No AND BL.Book\_Id=B.Book\_Id;

1. **For each library branch, retrieve the branch name and the total number of books loaned out from that branch.**

SELECT L.Branch\_Name, COUNT(\*)

FROM BOOKs\_LOANS B, LIBRARY\_BRANCH L

WHERE B.Branch\_Id = L.Branch\_Id

GROUP BY L.Branch\_Name

1. **Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.**

SELECT B.Name, B.Address, COUNT(\*)

FROM BORROWER B, BOOKs\_LOANS L

WHERE B.Card\_No = L.Card\_No

GROUP BY B.Card\_No

HAVING COUNT(\*) > 5;

1. **For each book authored (or co-authored) by "Salma", retrieve the title and the number of copies owned by the library branch whose name is "First branch".**

SELECT TItle, No\_Of\_Copies

FROM ( ( (BOOK\_AUTHORS NATURAL JOIN BOOK)

NATURAL JOIN BOOK\_COPIES)

NATURAL JOIN LIBRARY\_BRANCH)

WHERE Author\_Name = 'Salma' and Branch\_Name = 'First branch';

|  |  |
| --- | --- |
| 1. **Assessment** | *.* |

1. Outcome 1: Develop databases and associated queries using SQL **(SO:2; PI:2.2)**

|  |  |
| --- | --- |
| 1. **Resources** |  |

1. Natural Join: w3resource.com/mysql/advance-query-in-mysql/mysql-natural-join.php
2. Current date function: <https://www.w3schools.com/sql/func_mysql_curdate.asp>
3. Group By & Having clause: <https://www.guru99.com/group-by.html>
4. Inner vs Outer Join: <https://www.diffen.com/difference/Inner_Join_vs_Outer_Join#:~:text=Outer%20Join,-Diffen%20%E2%80%BA%20Technology%20%E2%80%BA%20Computers&text=In%20SQL%2C%20a%20join%20is,some%20dissimilar%20data%20from%20tables>.
5. Exists vs IN: [**http://www.dba-oracle.com/t\_exists\_clause\_vs\_in\_clause.htm**](http://www.dba-oracle.com/t_exists_clause_vs_in_clause.htm)