**Group B - Assignment No. 3**

**Aim:** To study & implement nested queries using set membership (in, not in), set comparison (<some, >=some, <all etc.) and set cardinality (unique, not unique) in SQL.

Theory :

### Set Operations (definition, syntax and examples)

1. Union

2. Intersect

3. Except(Minus)

### Nested Queries (Syntax and Examples)

* **set membership (IN,NOT IN)**
* **set comparisons (some, *<*= some, *>*= some, = some, and *<>* some** comparisons)
* **set cardinality(UNIQUE,Not UNIQUE)**

**Input:** Given schema of library.

**Schema**

**Book**---> {ISBNNO, Title, Edition, Price, Quantity,PubID(FK),AID(FK)}

**Author**---> {AuthorID, AName, Aemail}

**Publisher**---> {PID, Pub\_Name, PCity, PEmail}

**Student**---> { PRNNO, SName, RollNo, Branch, Year }

**BorrowedBy**---> {TRXID,IssueDate, RetDate, Fine, PRN(FK),ISBN(FK) }

**Execute following DML Statements.**

1. Display common names in Author Names and Student Names
2. Display names of Authors whose name is not in Student Name
3. Display student prnno & name who has issued book.
4. Display student prnno & name who has not issued book.
5. Display Book titles whose price is Less than some books published by publisher ‘Oxford’
   1. Using Tuple Variable
   2. Without Using Tuple Variable
6. Display book titles along with their publishers whose price is less than all books published by ‘Oxford’
7. Display book titles along with their publishers whose price is greater than some books published by ‘Oxford’

**Conclusion:** Studied different nested queries for Set Membership & comparisons.

**Group B - Assignment No. 4**

**Aim:** Write and execute suitable row level and statement level triggers.

**Theory:**

* **Introduction to SQL Trigger**
* **Uses for triggers :**
* **Types of triggers( with syntax and examples )**

There are two types of triggers based on the level it is triggered.

**1) Row level trigger** - An event is triggered for each row updated, inserted or deleted.

**2) Statement level trigger** - An event is triggered for each sql statement executed.

* **Trigger Execution Hierarchy**
* **Syntax for Trigger**
* **Example:** **Write a trigger which will restrict insertion of new employee record on Saturday & Sunday.**

**delimiter //**

**drop trigger if exists t1;**

**create trigger t1**

**before insert on employee**

**for each row**

**begin**

**if (Select DAYNAME(curdate()) in ('Saturday', 'Sunday'))then**

**signal SQLSTATE '02000' SET MESSAGE\_TEXT='YOU CAN NOT INSERT ON FRIDAY';**

**end if;**

**end;**

|  |
| --- |
|  |
|  |

* **Deleting a trigger (Syntax and example)**

**Example 2: write a trigger to update quantity of book if that book is issued to student**

**delimiter //**

**drop trigger if exists t2;**

**create trigger t2**

**after insert on BorrowedBy**

**for each row**

**begin**

**update Book**

**SET Quantity = Quantity – 1**

**where ISBNNO=NEW.ISBN;**

**end;**

**Q. Consider the Input:** Given schema of library.

**Schema**

**Book**---> {ISBNNO, Title, Edition, Price, Quantity,PubID(FK),AID(FK)}

**Author**---> {AuthorID, AName, Aemail}

**Publisher**---> {PID, Pub\_Name, PCity, PEmail}

**Student**---> { PRNNO, SName, RollNo, Branch, Year }

**BorrowedBy**---> {TRXID,IssueDate, RetDate, Fine, PRN(FK),ISBN(FK) }

**Write trigger for following**

1. Write a trigger to not allow librarian to issue books on Saturday & Sunday
2. Write a trigger to restrict students from entering lowercase values or extra spaces in their names .If entered, convert the same to Uppercase and also trim extra spaces.
3. Write a trigger to create a copy of one table after insertion.
4. Write a trigger to implement on delete cascade referential integrity for any of above tables.
5. Write a trigger to create a copy of old and new values for student’s year after update operation.
6. Write a trigger to create backup of whichever row will be deleted from any table from above database.

**CONCLUSION:** Studied and implemented SQL Triggers

**FAQs:-**

1) What is trigger?

2) What are the types of trigger?

3) How to delete a trigger?

4) What is row level and statement level trigger?

**Group B - Assignment No. 5**

**Aim** : Write and execute PL/SQL stored procedure and function to perform a suitable task on the database.Demonstrate its use.\

**Theory :**

* Difference between SQL and PL/SQL
* Introduction to PL/SQL
* PL/SQL constructs :
* Stored Procedures(Definition,use,Syntax and examples)
* Stored functions (Definition,Use,Syntax and examples)
* Difference between stored procedures and functions
* Calling stored procedures
* Calling stored functions
* Variable declaration
* Types of parameters in Stored procedures
* MySQL Programming constructs :
* Selection : IF,IF ELSE,ELSEIF Ladder,CASE statement with use,syntax and examples.
* Looping : While,Loop,Repeat Until,For etc,Use of Labels.

**a.Write a pl/sql stored procedure for the following**

Q1.To display the maximum values from given 3 values.

Q2.Display sum of numbers from 1 to 100 that are divisible by 4

Q3.Display the name of employee based on the given employee number.

Q4.Display the count of employees working in ‘IT’ department.

Q5.From given schema, Find no of books issued by every students using stored procedure

**b.Write PL/SQL stored functions for the following**

Q1.Display the factorial of given number

Q2.Display if given number is prime or not.

Q3.Display the number of employees working in department if dept id is given as argument.

Q4.Display the class based on percentage value.

Q5. From given schema , find total fine paid by student using stored function accept prnno as input.

**Group B - Assignment No. 6**

**Aim** : **Write a PL/SQL block to implement all types of cursors.**

**Theory :**

* Introduction to Cursors
* Types of Cursors
* Types of MySQL cursors
* Explicit Cursor(Syntax and detailed explanation with Example)
* Implicit cursor(Syntax and detailed explanation with example)

**Q.Write explicit cursors for following :**

**Consider Given Schema :**

***Emp(eno,ename,salary)***

1.Display the names of employees whose salary > 40000

2.Copy the contents of emp table to emp\_log tables.

3.Update the salary of all employees as per following conditions.

* If salary >= 70000 then increment salary by 5000
* If 50000<= salary< 75000 then increment salary by 3000
* Else increment salary by 1500.

And save the changes in a new table having eid and salary as the columns.

**Group B - Assignment No. 7**

**Aim : To study and implement the Database views.**

**Theory :**

* Introduction to Views
* Advantages of database views
* Syntax for creating Views with examples
* Example of DML operations on Views
* Dropping a view

**Q.Write MySQL queries for the following :**

**Consider Schema**

***Book****---> {ISBNNO, Title, Edition, Price, Quantity,PubID(FK),AID(FK)}*

***Author****---> {AuthorID, AName, Aemail}*

***Publisher****---> {PID, Pub\_Name, PCity, PEmail}*

***Student****---> { PRNNO, SName, RollNo, Branch, Year }*

***BorrowedBy****---> {TRXID,IssueDate, RetDate, Fine, PRN(FK),ISBN(FK) }*

* 1. Create view v1 which will contain prn no., student name & roll no. perform insert ,update & delete operation on view.
  2. Create a view v2 which will contain ISBN no, title, author id, author name. Try to insert few records in base table using view.

**Conclusion :**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF GROUP B\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*