K. Abhishek
TCOA76

Page No.:
Date:

Assignment No.: 3

Aim

Design at least 10 SQL queries for suitable database application using SQL DML statements: Insert, Select, Update, Delete with operators, functions, and set operator.

Objectives

To study SQL DML statements

Theory

Data Manipulation Language (DML)

It is a family of syntax elements similar to a computer programming language used for selecting, inserting, deleting and updating data in a database.
Performing read-only queries of data is sometimes also considered a component of DML.
DML language comprises the SQL data change statements, which modify stored data but not the schema or database object.

Inserting Data into Table:

To insert data in table, you would need to use SQL INSERT INTO command.

Syntax:

```
INSERT INTO table_name (field1, field2,... fieldN)
VALUES (value1, value2,... valueN);
```

Example:

```
INSERT INTO tut_table (tut_title, tut_author, submission_dt)
VALUES ("Java-SQL Tutorial", "Sam", "2021-11-20");
```

Fetching data from table

SQL SELECT command is used to fetch data from database.

Syntax:
```
SELECT field1, field2, ...
FROM table_name;
```

Example:
```
SELECT CustomerName
FROM Customers;
```

SQL UPDATE statement

It is used to modify the existing records in a table.

Syntax
```
UPDATE table_name
SET column1 = value1, column2=value2, ...
WHERE condition;
```

Example.
```
UPDATE Customers
SET ContactName= 'Alfreid', City= 'Frankfurt'
WHERE CustomerID=1;
```

Delete Query

If you want to delete a record from any table, then you can use SQL command DELETE FROM. You can use this command at prompt.

Syntax:
```
DELETE FROM table_name [WHERE clause]
```

- If WHERE clause is not specified, then all the records will be deleted from the table.

Example:
```
DELETE FROM tut_table WHERE tut_id=3;
```

○ Functions.

MySQL has many built-in functions.
There are many types of functions in MySQL such as string, numeric, date and some advanced function.

Example:
MySQL String Functions - CONCAT, LCASE, MID, SUBSTR, etc.
MySQL Numeric Functions - ASIN, EXP, LOG, MIN, etc.
MySQL Date Functions - CURTIME, DATE, DAY, EXTRACT, etc.
MySQL Advanced Functions - BIN, IF, NULLIF, USER, etc.

○ SET operator

It is a special type of operators which are used to combine the result of two queries.
Operators covered under SET operator are:

- Union
- Union All
- Intersect
- Minus

Rules for using SET operators in SQL:
- The number and order of columns must be same
- Data types must be compatible

Syntax:

- Union : SELECT * FROM table_name UNION SELECT * FROM table2;
- UnionAll : SELECT * FROM table1 UNION ALL SELECT * FROM table2;
- Intersect : SELECT * FROM table1 INTERSECT SELECT * FROM table2;
- Minus : SELECT * FROM table1 MINUS SELECT * FROM table2;

Conclusion

Thus, we have studied to use & implement various DML queries.

SQL> select * from account;

ACC_NO	BRANCH_NAME	BALANCE
1001	Akurdi	15000
1002	Nigdi	11000
1003	Chinchwad	20000
1004	Wakad	10000
1005	Akurdi	14000
1006	Nigdi	17000

6 rows selected.

SQL> select * from branch;

BRANCH_NAME	BRANCH_CITY	ASSETS
Akurdi	Pune	200000
Nigdi	Pimpri_chinchwad	300000
Wakad	Pune	100000
Chinchwad	Pimpri_chinchwad	400000
Sangvi	Pune	230000

SQL> select * from customer1;

CUST_NAME	CUST_STREET	CUST_CITY
Rutuja	JM road	Pune
Alka	Senapati road	Pune
Samiksha	Savedi road	Pimpri_chinchwad
Trupti	Lakshmi road	Pune
Mahima	Pipeline road	Pimpri_chinchwad
Ayushi	FC road	pune
Priti	Camp road	Pimri_chinchwad

7 rows selected.

SQL> select * from depositer;

CUST_NAME	ACC_NO
Rutuja	1005
Trupti	1002
Samiksha	1004

Loan(loan_no,branch_name,amount) :

SQL> select * from loan;

LOAN_NO	BRANCH_NAME	AMMOUNT
2001	Akurdi	2000
2002	Nigdi	1200
2003	Akurdi	1400
2004	Wakad	1350
2005	Chinchwad	1490
2006	Akurdi	12300
2007	Akurdi	14000

7 rows selected.

SQL> select * from borrower;

CUST_NAME	LOAN_NO
Mahima	2005
Trupti	2002
Rutuja	2004
Ayushi	2006
Priti	2007

SQL>select branch_name from loan;

```
BRANCH_NAME
--------------------
Akurdi
Nigdi
Akurdi
Wakad
Chinchwad
Akurdi
Akurdi

7 rows selected.
```

SQL> select loan_no from loan where branch_name='Akurdi' and amount>12000;

```
LOAN_NO
----------
2006
2007
```

SQL> select b.cust_name,b.loan_no,l.amount from borrower b inner join loan l on b.loan_no=l.loan_no;

CUST_NAME	LOAN_NO	AMOUNT
Trupti	2002	1200
Rutuja	2004	1350
Mahima	2005	1490
Ayushi	2006	12300
Priti	2007	14000

SQL> select b.cust_name from borrower b inner join loan l on b.loan_no=l.loan_no where l.branch_name='Akurdi'order by b.cust_name;

```
CUST_NAME
--------------------
Ayushi
Priti
```

SQL>select cust_name from depositer union select cust_name from borrower;

```
CUST_NAME
--------------------
Ayushi
MahimaPriti
Rutuja
Samiksha
Trupti

6 rows selected.
```

SQL> select cust_name from depositer intersect select cust_name from borrower;

CUST_NAME

Rutuja
Trupti

SQL> select cust_name from depositer minus select cust_name from borrower;

CUST_NAME

Samiksha

SQL> select avg(balance) from account where branch_name='Akurdi';

AVG(BALANCE)

14500

SQL> select branch_name,avg(balance) from account group by branch_name;

BRANCH_NAME	AVG(BALANCE)
Chinchwad	20000
Nigdi	14000
Wakad	10000
Akurdi	14500

SQL> select branch_name,count(branch_name) from account a inner join depositer d on a.acc_no=d.acc_no group by branch_name;

BRANCH_NAME	COUNT(BRANCH_NAME)
Nigdi	1
Wakad	1
Akurdi	1

SQL> select branch_name from account group by branch_name having avg(balance)>1200;

BRANCH_NAME

Chinchwad
Nigdi
Wakad
Akurdi

SQL> select count(cust_name) no_of_tuples from customer1;

NO_OF_TUPLES

7

SQL> select sum(amount) total_loan_amount from loan;

TOTAL_LOAN_AMOUNT

33740

SQL> delete from loan where amount>1300 and amount<1500;

LOAN_NO	BRANCH_NAME	AMOUNT
2001	Akurdi	2000
2002	Nigdi	1200
2006	Akurdi	12300
2007	Akurdi	14000

SQL>delete from branch where branch_name='Nigdi';

SQL> create public synonym cust2 for customer1;
Synonym created.