**EX.NO:2**

**DATA MANIPULATION LANGUAGE**

**DATE:**

**AIM:**

To perform operations in sql using Data Manipulation Language methods. The following queries are to be used.

**1) INSERT:**

**SYNTAX:**

insert into tablename values(value1,value2,value3,…….value n);

or

insert into tablename(col 1,col 2,…..col n)values(val 1,val 2,….val n);

or

insert into tablename values(&col 1,&col 2,&col 3,…..&col n)

**DESCRIPTION:**To insert records into the table.

**OUTPUT:**

Insert into student values(101,’srini’,98);

1 row inserted.

Or

Insert into student(rollno,name,marks)values(101,’srini’,98);

1 row inserted.

Or

Insert into student values(&rollno,’&name’,&marks);

Enter the rollno:101

Enter the name:srini

Enter the marks:98

1 row inserted.

**2) SELECT:**

**SYNTAX:**

Select \* from tablename;

**DESCRIPTION:**to view the entire records inserted into the table.

Select \* from tablename where columnname=value;

**DESCRIPTION:**to view the records of specific column.

**OUTPUT:**

Select \* from student;

ROLLNO NAME MARKS

101 srini 98

102 krish 85

103 vidhya 97

104 suganthi 78

4 row selected.

**3) UPDATE:**

**SYNTAX:**

Update tablename set columnname=values where columnname=values;

**DESCRIPTION:**to update or change the records in the table.

**OUTPUT:**

Update student set marks=100 where rollno=101;

1 row updated.

**4) DELETE:**

**SYNTAX:**

Delete from tablename;

Or

Truncate from tablename;

Or

Delete from tablename where columnname=values;

**DESCRIPTION:**to delete entire records in the table or to delete records of specific column.

**OUTPUT:**

Delete from student where rollno=104;

1 row deleted.

**4.1) AND/OR COMMAND:**

**SYNTAX:**

Select columnname from tablename where(condition 1 and condition 2);

Or

Select columnname from tablename where(condition 1 or condition 2);

**DESCRIPTION:**to view the records according to the given conditions.

**OUTPUT:**

Select rollno from student where (rollno>102 and rollno<104);

ROLLNO NAME MARKS

102 krish 85

103 vidhya 97

2 rows selected.

**4.2) BETWEEN:**

**SYNTAX:**

Select \* from tablename where columnname between values;

**DESCRIPTION:**to view the records using between command.

**OUTPUT:**

Select \* from student where rollno between 102 and 104;

ROLLNO NAME MARKS

103 vidhya 97

1 row selected.

**4.3) NOT BETWEEN:**

**SYNTAX:**

Select \* from tablename where columnname notbetween values;

**DESCRIPTION:** to view the records using not between command.

**OUTPUT:**

Select \* from student where rollno between 102 and 104;

ROLLNO NAME MARKS

101 srini 98

1 row selected.

**4.4) LIKE:**

**SYNTAX:**

Select \* from student where name like ‘k%’;

Or

Select \* from student where name like ‘%h’;

Or

Select \* from student where name like ‘\_ris%’;

**DESCRIPTION:**to view the row where name starts with ‘k’ or finish with ‘h’ or has letter ‘ris’.

**OUTPUT:**

ROLLNO NAME MARKS

102 krish 85

1 row selected.

**4.5) IN:**

**SYNTAX:**

Select \* from tablename where columnname in (values);

**DESCRIPTION:**to select the rows according to given values.

**OUTPUT:**

Select \* from student where marks in(98,78);

ROLLNO NAME MARKS

101 srini 98

104 suganthi 78

2 rows selected.

**4.6) ORDER BY:**

**SYNTAX:**

Select \* from student order by columnname asc or desc;

**DESCRIPTION:**to arrange the rows either in ascending or descending manner.

**OUTPUT:**

Select \* from student order by rollno desc;

ROLLNO NAME MARKS

104 suganthi 78

103 vidhya 97

102 krish 85

101 srini 98

4 rows selected.

**4.7) COUNT:**

**SYNTAX:**

Select count(columnname)from tablename;

**DESCRIPTION:**to view the number of columns present in the table.

**OUTPUT:**

Select count(rollno)from student;

Count(rollno)

4

4 rows selected.

**4.8) SUM:**

**SYNTAX:**

Select sum(columnname)from tablename;

**DESCRIPTION:**to view the sum of the values given in the table.

**OUTPUT:**

select sum(marks)from student;

sum(marks)

358

4 rows selected.

**4.9) MAX:**

**SYNTAX:**

Select max(columnname)from tablename;

**DESCRIPTION:**to view the maximum of the values given in the table.

**OUTPUT:**

select max(marks)from student;

max(marks)

98

1 row selected.

**4.10) AVG:**

**SYNTAX:**

Select avg(columnname)from tablename;

**DESCRIPTION:**to view the average of the values given in the table.

**OUTPUT:**

Select avg(marks)from student;

Avg(marks)

89

4 rows selected.

**4.11) ROUND:**

**SYNTAX:**

Select round(marks,2)from tablename;

**DESCRIPTION:**to round off the marks to two decimal points(if marks is created with float value).

**OUTPUT:**

Select round(marks,2)from student;  
round(marks)

98.56

85.48

97.57

78.85

4 rows selected.

**4.12) REVENUE:**

Create table with tablename product with columns id,price,units and multiply price and units and put it in a column revenue.

ID PRICE UNITS

101 71.75 9

102 60.75 7

**SYNTAX:**

Select id,price\*units revenue from product order by price\*units desc.

**DESCRIPTION:**to view id and revenue(product of price and units)columns in descending order.

**OUTPUT:**

ID REVENUE

102 645.75

101 471.51

2 rows selected.

**4.13) GROUPING:**

Create a table as tablename-store.

NAME SALES

Waves 11

Max 9

Waves 7

Max 10

**SYNTAX:**

Select storename,sum(sales) from tablename group by storename;

**DESCRIPTION:**to view the storename and sum of sales by grouping the name.

**OUTPUT:**

NAME SUM(SALES)

Waves 18

Max 19

4 rows selected.

**4.14) HAVING:**

**SYNTAX:**

Select name,sum(sales)from store having sum(sales)>20 group by name;

**DESCRIPTION:**to view the storename and sum of sales where sum of sales is greater than 20 and group by name.

**OUTPUT:**

NAME SUM(SALES)

Waves 18

Max 19

4 rows selected.

**4.15) CONCATENATE:**

**SYNTAX:**

Select concat(name,sales)from store;

**DESCRIPTION:**to concatenate the name and sales column from store table.

**OUTPUT:**

Concat(name,sales)

Waves11

Max9

Waves7

Max10

4 rows selected.

**4.16) SUBSTRING:**

**SYNTAX:**

Select substr(name,3)from store;

**DESCRIPTION:**to view the substring of names in 3 characters.

**OUTPUT:**

Substr

Ves

X

Ves

X

**4.17) UNION:**

The First table,

ID Name

1 abhi

2 adam

The Second table,

ID Name

2 adam

3 Chester

**SYNTAX:**

Select \* from FirstUNIONSelect \* from second;

**DESCRIPTION:**is used to combine the results of two or more Select statements

**OUTPUT:**

ID NAME

1 abhi

2 adam

3 Chester

**4.18) UNION ALL:**

**SYNTAX:**

Select \* from FirstUNION ALL Select \* from second;

**DESCRIPTION:**This operation is similar to Union. But it also shows the duplicate rows

**OUTPUT:**

ID NAME

1 abhi

2 adam

2 adam

3 Chester

**4.19) INTERSECT:**

**SYNTAX:**

Select \* from FirstINTERSECTselect \* from second;

**DESCRIPTION:**operation is used to combine two SELECT statements, but it only returns the records which are common from both SELECT statements

**OUTPUT:**

2 adam

**4.20) MINUS:**

**SYNTAX:**

Select \* from First MINUS select \* from second;

**DESCRIPTION:**Minus operation combines result of two Select statements and return only those result which belongs to first set of result

**OUTPUT:**

1 abhi

**4.21) LTRIM:**

**SYNTAX:**

SELECT LTRIM(‘ Sentence ‘);;

**DESCRIPTION:**Removes the spaces on the left side, or the trailing spaces from a string

**OUTPUT:**

‘Sentence ‘

**4.22) RTRIM:**

**SYNTAX:**

SELECT RTRIM(‘ Sentence ‘);;

**DESCRIPTION:R**emoves the spaces on the right side, or the leading spaces from a string

**OUTPUT:**

‘ Sentence‘

**5) JOIN:**

**5.1) INNER JOIN:**

"Orders" table

**OrderID CustomerID OrderDate**

**10308 1 1996-09-18**

**10309 2 1996-09-19**

**10310 3 1996-09-20**

**10311 3 1996-10-29**

"Customers" table

**CustomerID CustomerName ContactName Country**

**1 Alfreds Futterkiste Maria Anders Germany**

**2 Ana Trujillo Ana Trujillo Mexico**

**3 Antonio Taquería Antonio Moreno Mexico**

**SYNTAX:**

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate

FROM Orders

INNER JOIN Customers

ON Orders.CustomerID=Customers.CustomerID;

**DESCRIPTION:**is used to combine rows from two or more tables, based on a common field between them

**OUTPUT:**

OrderID CustomerName OrderDate

10308 **Alfreds Futterkiste** **1996-09-18**

10309 **Ana Trujillo**  **1996-09-19**

10310 **Antonio Taquería** **1996-09-20**

**5.2) LEFT JOIN:**

**SYNTAX:**

SELECT Orders.OrderID, Customers.CustomerName,Orders.OrderDateFROM OrdersLEFT JOIN CustomersON Orders.CustomerID=Customers.CustomerID;

**DESCRIPTION:**The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2)

**OUTPUT:**

OrderID CustomerName OrderDate

10308 **Alfreds Futterkiste** **1996-09-18**

10309 **Ana Trujillo**  **1996-09-19**

10310 **Antonio Taquería** **1996-09-20**

10311 **Antonio Taquería** **1996-10-29**

**5.3) RIGHT JOIN:**

**SYNTAX:**

SELECT Orders.OrderID, Customers.CustomerName,Orders.OrderDate FROM Orders RIGHT JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

**DESCRIPTION:**The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1)

**OUTPUT:**

OrderID CustomerName OrderDate

10308 **Alfreds Futterkiste** **1996-09-18**

10309 **Ana Trujillo**  **1996-09-19**

10310 **Antonio Taquería**  **1996-09-20**

**5.4) FULL OUTER JOIN:**

**SYNTAX:**

SELECT Orders.OrderID, Customers.CustomerName,Orders.OrderDate FROM Orders FULL OUTER JOINCustomers ON Orders.CustomerID=Customers.CustomerID;

**DESCRIPTION:**The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2)

**OUTPUT:**

OrderID CustomerName OrderDate

10308 **Alfreds Futterkiste** **1996-09-18**

10309 **Ana Trujillo**  **1996-09-19**

10310 **Antonio Taquería** **1996-09-20**

**10311 1996-10-29**

**6) DATE FUNCTIONS:**

**6.1) NOW(),CURDATE(),CURTIME()**

**SYNTAX:**

CREATE TABLE Orders

(OrderId int NOT NULL,

ProductName varchar(50) NOT NULL,

OrderDate datetime NOT NULL DEFAULT NOW(),

PRIMARY KEY (OrderId))

**DESCRIPTION:**NOW() returns the current date and time

**OUTPUT:**

OrderId ProductName OrderDate

1. Jarlsberg Cheese 2014-11-11 13:23:44.657

**6.2) DATE()**

OrderId ProductName OrderDate

1. Jarlsberg Cheese 2014-11-11 13:23:44.657

**SYNTAX:**

SELECT ProductName, DATE(OrderDate) AS OrderDate

FROM Orders

WHERE OrderId=1

**DESCRIPTION:**The DATE() function extracts the date part of a date or date/time expression

**OUTPUT:**

roductName OrderDate

Jarlsberg Cheese 2014-11-11

**6.3) EXTRACT()**

OrderId ProductName OrderDate

1. Jarlsberg Cheese 2014-11-11 13:23:44.657

**SYNTAX:**

SELECT EXTRACT(YEAR FROM OrderDate) AS OrderYear,

EXTRACT(MONTH FROM OrderDate) AS OrderMonth,

EXTRACT(DAY FROM OrderDate) AS OrderDay,

FROM Orders

WHERE OrderId=1

**DESCRIPTION:**The EXTRACT() function is used to return a single part of a date/time, such as year, month, day, hour, minute, etc.

Where date is a valid date expression and unit can be one of the following:

Unit Value

MICROSECOND

SECOND

MINUTE

HOUR

DAY

WEEK

MONTH

QUARTER

YEAR

SECOND\_MICROSECOND

MINUTE\_MICROSECOND

MINUTE\_SECOND

HOUR\_MICROSECOND

HOUR\_SECOND

HOUR\_MINUTE

DAY\_MICROSECOND

DAY\_SECOND

DAY\_MINUTE

DAY\_HOUR

YEAR\_MONTH

**OUTPUT:**

OrderYear OrderMonth OrderDay

2014 11 11

**6.4) DATE\_ADD()**

OrderId ProductName OrderDate

1. Jarlsberg Cheese 2014-11-11 13:23:44.657

**SYNTAX:**

SELECT OrderId,DATE\_ADD(OrderDate,INTERVAL 45 DAY) AS OrderPayDate

FROM Orders

**DESCRIPTION:**The DATE\_ADD() function adds a specified time interval to a date.

Where date is a valid date expression and expr is the number of interval you want to add.

type can be one of the following:

Type Value

MICROSECOND

SECOND

MINUTE

HOUR

DAY

WEEK

MONTH

QUARTER

YEAR

SECOND\_MICROSECOND

MINUTE\_MICROSECOND

MINUTE\_SECOND

HOUR\_MICROSECOND

HOUR\_SECOND

HOUR\_MINUTE

DAY\_MICROSECOND

DAY\_SECOND

DAY\_MINUTE

DAY\_HOUR

YEAR\_MONTH

**OUTPUT:**

OrderId OrderPayDate

1. 2014-12-26 13:23:44.657

**6.5) DATE\_SUB()**

OrderId ProductName OrderDate

1. Jarlsberg Cheese 2014-11-11 13:23:44.657

**SYNTAX:**

SELECT OrderId,DATE\_SUB(OrderDate,INTERVAL 5 DAY) AS SubtractDate

FROM Orders

**DESCRIPTION:**The DATE\_SUB() function subtracts a specified time interval from a date.

Where date is a valid date expression and expr is the number of interval you want to add.

type can be one of the following:

Type Value

MICROSECOND

SECOND

MINUTE

HOUR

DAY

WEEK

MONTH

QUARTER

YEAR

SECOND\_MICROSECOND

MINUTE\_MICROSECOND

MINUTE\_SECOND

HOUR\_MICROSECOND

HOUR\_SECOND

HOUR\_MINUTE

DAY\_MICROSECOND

DAY\_SECOND

DAY\_MINUTE

DAY\_HOUR

YEAR\_MONTH

**OUTPUT:**

OrderId SubtractDate

1. 2014-11-06 13:23:44.657

**6.6) DATEDIFF ()**

**SYNTAX:**

SELECT DATEDIFF('2014-11-30','2014-11-29') AS DiffDate

**DESCRIPTION:**The DATEDIFF() function returns the time between two dates.

**OUTPUT:**

DateDiff

1

**6.7) DATE\_FORMAT()**

**SYNTAX:**

DATE\_FORMAT(NOW(),'%b %d %Y %h:%i %p')

DATE\_FORMAT(NOW(),'%m-%d-%Y')

DATE\_FORMAT(NOW(),'%d %b %y')

DATE\_FORMAT(NOW(),'%d %b %Y %T:%f')

**DESCRIPTION:**The DATE\_FORMAT() function is used to display date/time data in different formats.

Where date is a valid date and format specifies the output format for the date/time.

The formats that can be used are:

Format Description

%a Abbreviated weekday name

%b Abbreviated month name

%c Month, numeric

%D Day of month with English suffix

%d Day of month, numeric (00-31)

%e Day of month, numeric (0-31)

%f Microseconds

%H Hour (00-23)

%h Hour (01-12)

%I Hour (01-12)

%i Minutes, numeric (00-59)

%j Day of year (001-366)

%k Hour (0-23)

%l Hour (1-12)

%M Month name

%m Month, numeric (00-12)

%p AM or PM

%r Time, 12-hour (hh:mm:ss AM or PM)

%S Seconds (00-59)

%s Seconds (00-59)

%T Time, 24-hour (hh:mm:ss)

%U Week (00-53) where Sunday is the first day of week

%u Week (00-53) where Monday is the first day of week

%V Week (01-53) where Sunday is the first day of week, used with %X

%v Week (01-53) where Monday is the first day of week, used with %x

%W Weekday name

%w Day of the week (0=Sunday, 6=Saturday)

%X Year for the week where Sunday is the first day of week, four digits, used with %V

%x Year for the week where Monday is the first day of week, four digits, used with %v

%Y Year, four digits

%y Year, two digits

**OUTPUT:**

Nov 04 2014 11:45 PM

11-04-2014

04 Nov 14

04 Nov 2014 11:45:34:243

**7) nth\_appearance**

**SYNTAX:**

INSTR( string, substring [, start\_position [, nth\_appearance ] ] )

**DESCRIPTION:**The nth appearance of substring.

**7.1) LENGTH**

**SYNTAX:**

SELECT LENGTH('CANDIDE') "Length in characters"

FROM DUAL;

**DESCRIPTION:**The LENGTH functions return the length of char. LENGTH calculates length using characters as defined by the input character set.

**OUTPUT:**

Length in characters7

**7.2) SUBQUERIES**

**SYNTAX:**

SELECT \* FROM Order WHERE Orders.OrderIDIN (SELECT Customers.CustomerID FROM Customerscols WHERE Customers.CustomerID= 1);

**DESCRIPTION:**a subquery is a query within a query. You can create subqueries within your SQL statements. These subqueries can reside in the WHERE clause, the FROM clause, or the SELECT clause.

**OUTPUT:**

OrderID CustomerName OrderDate

10308 **Alfreds Futterkiste** **1996-09-18**

**7.3) CAST**

**SYNTAX:**

SELECT CAST (OrderIDAS INT)FROM Order

**DESCRIPTION:**The CAST function converts a value from one data type to another and provides a data type to a dynamic parameter (?) or a NULL value.

**OUTPUT:**

OrderId is converted into INT