Practical -5
MySQL Built in Functions and Modification of Data

MySQL UPDATE

UPDATE statement is used to update existing data in a table. We can use the UPDATE statement to change column values of a single row, a group of rows, or all rows in a table

Syntax : UPDATE tablename
SET Column1=expr1 ,
Column2 = expr2, ...
WHERE condition

e.g : mysql> update employee set city='mumbai' where name='virat';

update multiple fileld value of row.
mysql> update employee set city='mumbai', joindate='2011-09-27' where name='sameer';

updating multiple rows who satisfy the condition
mysql> update employee set salary = salary+200 where age < 32 and department='production';

updating a filed which having primary key which refer in another table as foreign key

step 1 . try to update primary key filed
update category set cat_id=55 where cat_id=5;
error... at cat_id used as foreign key in post table.
step 2 , drop the foreign key constraint of post table.
alter table post drop constraint post_fk;
step 3 alter table with add new constraint at post table.
alter table post add constraint post_fk foreign key(id) references category(cat_id) on delete cascade on update cascade;
step 4 . execute update query on category table.
mysql > update category set cat_id=55 where cat_id=5;
mysql> select * from post;

it will update the child table relevant data also.

MySQL DELETE

To delete data from a table, you use the MySQL DELETE statement

Syntax : DELETE FROM tablename
WHERE condition;

Mysql> delete from books ; // delete all the records from book table.

Mysql > delete from post where id=5;

Deleteing record which is refer in another table as foreign key

step 1 . try to update primary key filed

delete from category set where cat_id=5;

error… at cat_id used as foreign key in post table.

step 2 , drop the foreign key constraint of post table.

alter table post drop constraint post_fk;

step 3 alter table with add new constraint at post table.

alter table post add constraint post_fk foreign key(id) references category(cat_id) on delete cascade on update cascade;

step 4 . execute delete query on category table.

Mysql> delete from category where cat_id = 5;

It will also delete all the records from child table post which contains id =5;

- **String Function.**

- **LENGTH() function**

  The LENGTH() function returns the length of a string (in bytes)

  LENGTH(STRING)

  e.g **select length('welcome to mysql') as lengthofstr;**

  lengthofstr |

  +----------+

  16

- **LEFT()**

  The LEFT() function extracts a number of characters from a string (starting from left)

  LEFT(STRING , noofchar)

  e.g **mysql> select left('welcome to mysql',4) ;**

  welc

  e.g mysql>select left(name,5) from employee;

  | Rohan |

  | virat |

  | samee | **extract first 5 character for name field for each row.**

  | Hares

- RIGHT()

  Extract 4 characters from a string (starting from right), from last .

  RIGHT(string,noofchar)

  e.g last 4 character of string

  mysql> select right('welcome to mysql',4) ;

  ysql

- **LOWER()**

  Convert string into lower case.

  LOWER(string)

  e.g **mysql> select LOWER('MYSQL') as lowstr**

mysql

- **UPPER()**
  Converts the string into uppercase
  UPPER(string)

  e.g mysql> select upper(name) from employee;
  VIRAT |
  | SAMEER |
  | HARES|

- **POSITION()**
  The POSITION() function returns the position of the first occurrence of a substring in a string.
  If the substring is not found within the original string, this function returns 0.
  This function performs a case-insensitive search.

  POSITION(substring IN string)

  e.g **mysql> select position('NEW' in 'Time for new year');**
  10

- **SUBSTRING()**
  The SUBSTRING() function extracts a substring from a string (starting at any position).
  SUBSTRING(*string*, *start*, *length*)
  String :Required. The string to extract from
  start :Required. The start position. Can be both a positive or negative number. If it is a positive number, this function extracts from the beginning of the string. If it is a negative number, this function extracts from the end of the string
  length : Optional. The number of characters to extract. If omitted, the whole string will be returned (from the start position)

  e.g **mysql> select substring('testing for substring',5,9);**
  ing for s

  mysql> select substring('testing for substring',5);
  ing for substring

  mysql> select substring('firstname lastname',1 , position(' ' in 'firstname lastname'));
  firstname

  fetch only first word from name field
  mysql> select substring(name,1 , position(' ' in name)) from employee;

  mysql> select substring('firstname lastname', -3);
  ame

  mysql> select substring('firstname lastname', -3,5);
  ame

- **REPLACE()**
  The REPLACE() function replaces all occurrences of a substring within a string, with a new substring.
  REPLACE(*string*, *from_string*, *new_string*)
  string :Required. The original string
  from_string : Required. The substring to be replaced
  new_string Required. The new replacement substring.

e.**g mysql> select replace('Thhis is new string','i','newi'); every I is replace with newi Thhnewis newis new strnewing**

- **CONCATE()**
  function adds two or more expressions together.
  CONCAT(*expression1*, *expression2*, *expression3*,...)

  mysql> select concat("welcome" , "new year") as str;
  +----------------+
  | str |
  +----------------+
  | welcomenew year |

- **Numeric functions**

- **MIN()**
  The MIN() function returns the minimum value in a set of values.
  MIN(expression)

  e.g display the minimum salary of employee
  mysql> select min(salary) from employee;
  5000

- **Max()**
  The MAX() function returns the maximum value in a set of values.
  MAX(*expression*)

  mysql> select name, max(salary) from employee;
  Rohan patel | 30000.00 |

- **POW()/POWER()**
  The POW() function returns the value of a number raised to the power of another number.
  POW(*base*, exponent)

  mysql> select pow(8,3);
  512

- **DATE Functions**

- **CURDATE()**
  Return the current date:
  e.g select CURDATE();

- **CURTIME()**
  Return the current Time:
  Mysql> Select curtime();
- **Now()**
  Returns current date and time both
  Mysql>Select now()

- **ADDDATE()**
  dds a time/date interval to a date and then returns the date.
  ADDDATE(*date*, INTERVAL *value addunit*)
  Or
  ADDDATE(*date*, *days*)

  date Required. The date to be modified
  days Required. The number of days to add to date
  value Required. The value of the time/date interval to add. Both positive and negative values are allowed
  addunit Required. The type of interval to add. Can be one of the following values:
  SECOND
  MINUTE
  HOUR
  DAY
  MONTH
  YEAR

e.g **mysql> select adddate(now(), interval 15 minute);**

mysql> select adddate(now(),interval 1 month); // add one month in date

mysql> select adddate(now(),interval -3 year); // subtract 3 year from current date

mysql> select adddate(now(),interval -3 day); //subtract 3 days from current date

- **DATEDIFF()**
  DATEDIFF(date1, date2)
date1, date2 Required. Two dates to calculate the number of days between.
(date1 - date2) answer will be difference of number of days
mysql> select datediff(now(),joindate) as difference from employee;

- **DATE_FORMAT()**
  DATE_FORMAT(*date*, *format*)
  date Required. The date to be formatted
  format Required. The format to use.

%a Abbreviated weekday name (Sun to Sat)

%b Abbreviated month name (Jan to Dec)
%D Day of the month as a numeric value, followed by suffix (1st, 2nd, 3rd, ...)
%d Day of the month as a numeric value (01 to 31)
%e Day of the month as a numeric value (0 to 31)
%f Microseconds (000000 to 999999)
%H Hour (00 to 23)
%h Hour (00 to 12)
%i Minutes (00 to 59)
%j Day of the year (001 to 366)
%k Hour (0 to 23)
%l Hour (1 to 12)
%M Month name in full (January to December)
%m Month name as a numeric value (00 to 12)
%p AM or PM
%r Time in 12 hour AM or PM format (hh:mm:ss AM/PM)
%S Seconds (00 to 59)
%s Seconds (00 to 59)
%T Time in 24 hour format (hh:mm:ss)
%U Week where Sunday is the first day of the week (00 to 53)
%u Week where Monday is the first day of the week (00 to 53)
%V Week where Sunday is the first day of the week (01 to 53). Used with %X
%v Week where Monday is the first day of the week (01 to 53). Used with %x
%W Weekday name in full (Sunday to Saturday)
%w Day of the week where Sunday=0 and Saturday=6
%X Year for the week where Sunday is the first day of the week. Used with %V
%x Year for the week where Monday is the first day of the week. Used with %v
%Y Year as a numeric, 4-digit value
%y Year as a numeric, 2-digit value

e.g mysql> select date_format(now(),'%d %b %Y');
22 Dec 2020

mysql> select date_format(now(),'%a %b %Y');
Tue Dec 2020

mysql> select date_format(now(),'%d /%m /%y');
22 /12 /20

EXERCISE

1. Add new field email id varchar(50) in employee table.
2. Update the city to anand and join date 2009-10-12 of employee rohan patel.
3. Add the 2 yr to existing age of employee who are worker.
4. Update the email id of every employee by using where condition on name.
5. Update email id to Null where designation is peon.
6. Update the city of 'reema' from surat to Mumbai .
7. Increase the salary of all employee of sales department by 500;
8. Create a copy of employee table name as office
   mysql> create table office like employee;
   insert record of employee table into office table.
9. Update the joindate with current date of employee suhana in office table.

10. Delete the records of employee where joindate=NULL from office table.
11. Delete the records of employee where age is >35 from office table.
12. Delete all records from the employee table.
13. Display the joindate in dd/mm/yy format.
14. Display the joindate of employee in format line day-mon-year e.g 12-aug-1999
15. Display the Joindate of employee like Tuesday 27th September 2011 format
16. Display the day of the year for the current date.
17. Find out the number of days of joining(date difference from current date) of employee working in admin department.
18. Display the employee name and designation in format name**designation.(use contact)
19. Replace '-' with '/' in employee joindate (replace is just for display purpose it does not change internal storage structure)

Queries based on practical -3 tables(order database)

22. Change the city of ClientNO 'C00005' to 'Bangalore'
23. Change the BalDue of ClientNo 'C00001' to Rs. 1000
24. Change the cost price of 'Trousers' to Rs. 950.00
25. Change the city of the salesman to Pune .
26. Change qtyonhand of 'skirt' to 80.
27. Increase the orderquanty by 1 of all orders for product P00001 in sales_order_detail table.