String Functions:

String manipulation functions are amongst the most-used functions in programming.

Selected MySQL string functions.

Function	**Description**
CONCAT	Concatenates data from two different character columns and returns a single
column	Syntax: CONCAT (strg_value, strg_value) UPPER/LOWER Returns a string in all capital or all lowercase letters Syntax: UPPER(strg_value) , LOWER(strg_value)
SUBSTR	Returns a substring or part of a given string parameter Syntax: SUBSTR (strg_value, p, l) where p = start position and l = length of characters
LENGTH	Returns the number of characters in a string value Syntax: LENGTH(strg_value)

SELECT CONCAT(EMP_LNAME ,EMP_FNAME) AS NAME FROM EMPLOYEE;

```
mysql> SELECT CONCAT(EMP_LNAME ,EMP_FNAME) AS
    -> NAME FROM EMPLOYEE;
+----------------+
| NAME           |
+----------------+
| CalderdaleEmma |
| RicardoMarshel |
| ArshadArif     |
| RobertsAnne    |
| DenverEnrica   |
| NamowaMirrelle |
| SmithGemma     |
+----------------+
7 rows in set (0.01 sec)
```

UPPER/LOWER:

Concatenates data from two different character columns and returns a single

SELECT CONCAT(UPPER(EMP_LNAME),LOWER(EMP_FNAME)) AS NAME FROM EMPLOYEE;

```
mysql> SELECT CONCAT(UPPER(EMP_LNAME),LOWER(EMP_FNAME)) AS
    -> NAME FROM EMPLOYEE;
+----------------+
| NAME           |
+----------------+
| CALDERDALEemma |
| RICARDOmarshel |
| ARSHADarif     |
| ROBERTSanne    |
| DENVERenrica   |
| NAMOWAmirrelle |
| SMITHgemma     |
+----------------+
7 rows in set (0.01 sec)
```

SUBSTR:

The following example lists the first three characters of all the employees' first name.

SELECT EMP_PHONE, SUBSTR(EMP_FNAME,1,3) FROM EMPLOYEE;

```
mysql> SELECT EMP_PHONE, SUBSTR(EMP_FNAME,1,3)
    -> FROM EMPLOYEE;
+-----------+-----------------------+
| EMP_PHONE | SUBSTR(EMP_FNAME,1,3) |
+-----------+-----------------------+
| 324-9134  | Emm                   |
| 324-4472  | Mar                   |
| 675-8993  | Ari                   |
| 898-3456  | Ann                   |
| 504-4434  | Enr                   |
| 890-3243  | Mir                   |
| 324-7845  | Gem                   |
+-----------+-----------------------+
7 rows in set (0.02 sec)
```

LENGTH:

The following example lists all attraction names and the length of their names; ordered descended by attraction name length.

SELECT ATTRACT_NAME, LENGTH(ATTRACT_NAME) AS NAMESIZE FROM ATTRACTION ORDER BY NAMESIZE DESC;

```
mysql> SELECT ATTRACT_NAME, LENGTH(ATTRACT_NAME) AS
    -> NAMESIZE FROM ATTRACTION
    -> ORDER BY NAMESIZE DESC;
+-----------------+----------+
| ATTRACT_NAME    | NAMESIZE |
+-----------------+----------+
| SpinningTeacups |       15 |
| ThunderCoaster  |       14 |
| FlightToStars   |       13 |
| 3D-Lego_Show    |       12 |
| UnderSeaWord    |       12 |
| BlackHole2      |       10 |
| Ant-Trap        |        8 |
| Carnival        |        8 |
| GoldRush        |        8 |
| Pirates         |        7 |
| NULL            |     NULL |
+-----------------+----------+
11 rows in set (0.10 sec)
```

Conversion Functions:

Conversion functions allow you to take a value of a given data type and convert it to the equivalent value in another data type. In MySQL, some conversions occur implicitly. For example, MySQL automatically converts numbers to strings when needed, and vice versa.

So if you enter the following query:

SELECT 10 + '10'

MySQL would give you an answer of 20 as it would automatically convert the string containing '10' into the number 10

SELECT 10 + "10";

```
mysql> SELECT 10 + "10";
+-----------+
| 10 + "10" |
+-----------+
|        20 |
+-----------+
1 row in set (0.04 sec)
```

CAST:

If you want to explicitly convert a number to a string, then you can use either the CAST or CONVERT function. However, MySQL 5.0 recommends only the CAST function is used.

Let's look at an example.

SELECT 10, CAST(10 AS CHAR);

```
mysql> SELECT 10, CAST(10 AS CHAR);
+----+------------------+
| 10 | CAST(10 AS CHAR) |
+----+------------------+
| 10 | 10               |
+----+------------------+
1 row in set (0.02 sec)
```

CASE:

The CASE function compares an attribute or expression with a series of values and returns an associated value or a default value if no match is found. There are two versions of the CASE function. The syntax of each is shown below.

SELECT PARK_CODE, PARK_COUNTRY, (CASE PARK_COUNTRY WHEN 'UK' THEN 'United Kingdom' WHEN 'FR' THEN 'France' WHEN 'NL' THEN 'The Netherlands' WHEN 'SP' THEN 'Spain' WHEN 'ZA' THEN 'South Africa' WHEN 'SW' THEN 'Switzerland' ELSE' Unknown' END) AS COUNTRY FROM THEMEPARK;

```
mysql> SELECT PARK_CODE, PARK_COUNTRY, (CASE PARK_COUNTRY WHEN 'UK'
    -> THEN 'United Kingdom' WHEN 'FR' THEN 'France' WHEN 'NL' THEN 'The
    '> Netherlands' WHEN 'SP' THEN 'Spain' WHEN 'ZA' THEN 'South Africa' WHEN 'SW'
    -> THEN 'Switzerland' ELSE' Unknown' END) AS COUNTRY
    -> FROM THEMEPARK;
+-----------+--------------+-----------------+
| PARK_CODE | PARK_COUNTRY | COUNTRY         |
+-----------+--------------+-----------------+
| FR1001    | FR           | France          |
| NL1202    | NL           | The
Netherlands |
| SP4533    | SP           | Spain           |
| SW2323    | SW           | Switzerland     |
| UK2622    | UK           | United Kingdom  |
| UK3452    | UK           | United Kingdom  |
| ZA1342    | ZA           | South Africa    |
+-----------+--------------+-----------------+
7 rows in set (0.01 sec)
```

Exercises

Exercise:

Create the view EMPFR and update the Theme Park that employee number 101 works in. (Update the employee number 101 information in the EMPFR view).

CREATE VIEW EMPFR AS

SELECT *

FROM EMPLOYEE

where park_code="UK3452"

with check option;

```
mysql> CREATE VIEW EMPFR AS
    -> SELECT *
    -> FROM EMPLOYEE
    -> where park_code="UK3452"
    -> with check option;
Query OK, 0 rows affected (0.11 sec)

mysql> select * from empfr;
+---------+-----------+-----------+-----------+------------+--------------+---------------+-----------+-----------+
| EMP_NUM | EMP_TITLE | EMP_LNAME | EMP_FNAME | EMP_DOB    | EMP_HIRE_DATE | EMP_AREA_CODE | EMP_PHONE | PARK_CODE |
+---------+-----------+-----------+-----------+------------+--------------+---------------+-----------+-----------+
|     101 | Ms        | Ricardo   | Marshel   | 1978-03-19 | 1996-04-25    | 0181          | 324-4472  | UK3452    |
|     103 | Ms        | Roberts   | Anne      | 1974-10-16 | 1994-08-16    | 0181          | 898-3456  | UK3452    |
+---------+-----------+-----------+-----------+------------+--------------+---------------+-----------+-----------+
2 rows in set (0.01 sec)
```

UPDATING emp_num=101 information

update empfr

set emp_lname="babar", emp_fname="ibrar",

emp_DOB="2000-01-23",emp_area_code=0182,

emp_phone="333-2132"

where emp_num=101;

```
mysql> update empfr
    ->      set emp_lname="babar", emp_fname="ibrar",
    -> emp_DOB="2000-01-23",emp_area_code=0182,
    -> emp_phone="333-2132"
    ->      where emp_num=101;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from empfr;
+---------+-----------+-----------+-----------+------------+--------------+---------------+-----------+-----------+
| EMP_NUM | EMP_TITLE | EMP_LNAME | EMP_FNAME | EMP_DOB    | EMP_HIRE_DATE | EMP_AREA_CODE | EMP_PHONE | PARK_CODE |
+---------+-----------+-----------+-----------+------------+--------------+---------------+-----------+-----------+
|     101 | Ms        | babar     | ibrar     | 2000-01-23 | 1996-04-25    | 182           | 333-2132  | UK3452    |
|     103 | Ms        | Roberts   | Anne      | 1974-10-16 | 1994-08-16    | 0181          | 898-3456  | UK3452    |
+---------+-----------+-----------+-----------+------------+--------------+---------------+-----------+-----------+
2 rows in set (0.00 sec)
```

Exercise:

Employee Emma Cauderdale (EMP_NUM =100) has now changed her phone number to

324-9652. Update her information in the EMPFR view. Write a query to show her new phone

number has been updated and then Remove the EMPFR view.

update EMPFR

set emp_phone="324-9652"

where emp_num=100;

```
mysql> update EMPFR
    -> set emp_phone="324-9652"
    -> where emp_num=100;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from empfr;
+---------+-----------+------------+-----------+------------+---------------+---------------+-----------+-----------+
| EMP_NUM | EMP_TITLE | EMP_LNAME  | EMP_FNAME | EMP_DOB    | EMP_HIRE_DATE | EMP_AREA_CODE | EMP_PHONE | PARK_CODE |
+---------+-----------+------------+-----------+------------+---------------+---------------+-----------+-----------+
|     100 | Ms        | Calderdale | Emma      | 1972-06-15 | 1992-03-15    | 0181          | 324-9652  | FR1001    |
|     102 | Mr        | Arshad     | Arif      | 1969-11-14 | 1990-12-20    | 7253          | 675-8993  | FR1001    |
|     105 | Ms        | Namowa     | Mirrelle  | 1990-03-14 | 2006-11-08    | 0181          | 890-3243  | FR1001    |
+---------+-----------+------------+-----------+------------+---------------+---------------+-----------+-----------+
3 rows in set (0.00 sec)

mysql> drop view empfr;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Exercise:

Create a view of only those Theme Parks where tickets have been sold and then display thecontents of this view.

Create view tparkssold as

select park_code,park_name,park_city,park_country

from sales join sales_line using (transaction_no)

join themepark using (park_code) group by (park_code);

```
mysql> Create view tparkssold as
    -> select park_code,park_name,park_city,park_country
    ->   from sales join sales_line using (transaction_no)
    ->  join themepark using (park_code) group by (park_code);
Query OK, 0 rows affected (0.05 sec)

mysql> select * from tparkssold;
+-----------+--------------+--------------+--------------+
| park_code | park_name    | park_city    | park_country |
+-----------+--------------+--------------+--------------+
| FR1001    | FairyLand    | PARIS        | FR           |
| UK3452    | PleasureLand | STOKE        | UK           |
| ZA1342    | GoldTown     | JOHANNESBURG | ZA           |
+-----------+--------------+--------------+--------------+
3 rows in set (0.00 sec)
```

Exercise:

The Theme Park managers want to create a view called EMP_DETAILS which contains the following information. EMP_NO, PARK_CODE, PARK_NAME, EMP_LNAME_EMP_FNAME, EMP_HIRE_DATE and EMP_DOB. The view should only be read only. Check that the view works, by displaying its contents.

CREATE view EMP_DETAIL AS

select T.PARK_CODE, T.PARK_NAME,

E.EMP_NUM, E.EMP_LNAME,E.EMP_FNAME, E.EMP_HIRE_DATE,E.EMP_DOB

FROM THEMEPARK T NATURAL JOIN EMPLOYEE E;

```
mysql> CREATE view EMP_DETAIL AS
    ->     select T.PARK_CODE, T.PARK_NAME,
    ->      E.EMP_NUM, E.EMP_LNAME,E.EMP_FNAME, E.EMP_HIRE_DATE,E.EMP_DOB
    ->      FROM THEMEPARK T NATURAL JOIN EMPLOYEE E;
Query OK, 0 rows affected (0.05 sec)

mysql> select * from emp_detail;
+-----------+-------------+---------+------------+-----------+---------------+------------+
| PARK_CODE | PARK_NAME   | EMP_NUM | EMP_LNAME  | EMP_FNAME | EMP_HIRE_DATE | EMP_DOB    |
+-----------+-------------+---------+------------+-----------+---------------+------------+
| FR1001    | FairyLand   |     100 | Calderdale | Emma      | 1992-03-15    | 1972-06-15 |
| UK3452    | PleasureLand|     101 | babar      | ibrar     | 1996-04-25    | 2000-01-23 |
| FR1001    | FairyLand   |     102 | Arshad     | Arif      | 1990-12-20    | 1969-11-14 |
| UK3452    | PleasureLand|     103 | Roberts    | Anne      | 1994-08-16    | 1974-10-16 |
| ZA1342    | GoldTown    |     104 | Denver     | Enrica    | 2001-10-20    | 1980-11-08 |
| FR1001    | FairyLand   |     105 | Namowa     | Mirrelle  | 2006-11-08    | 1990-03-14 |
| ZA1342    | GoldTown    |     106 | Smith      | Gemma     | 1989-01-05    | 1968-02-12 |
+-----------+-------------+---------+------------+-----------+---------------+------------+
7 rows in set (0.01 sec)
```

Exercise:

Using your view EMP_DETAILS, write a query that displays all employee first and last names and the park names.

select EMP_FNAME,EMP_LNAME,PARK_CODE FROM EMP_DETAIL;

```
mysql> select EMP_FNAME,EMP_LNAME,PARK_CODE FROM EMP_DETAIL;
+-----------+------------+-----------+
| EMP_FNAME | EMP_LNAME  | PARK_CODE |
+-----------+------------+-----------+
| Emma      | Calderdale | FR1001    |
| ibrar     | babar      | UK3452    |
| Arif      | Arshad     | FR1001    |
| Anne      | Roberts    | UK3452    |
| Enrica    | Denver     | ZA1342    |
| Mirrelle  | Namowa     | FR1001    |
| Gemma     | Smith      | ZA1342    |
+-----------+------------+-----------+
7 rows in set (0.00 sec)
```

Exercise:

Create a view called TICKET_SALES which contains details of the min, max and average sales at each Theme Park. The name of the theme park should also be displayed. (Hint 1: you will need to join three tables). Once you have created your view, write a query to display the contents.

Create view TICKET_SALES as

select park_name, min(line_price),max(line_price),avg(line_price)

from sales join sales_line using (transaction_no)

join themepark using (park_code) group by (park_code);

```
mysql> Create view TICKET_SALES as
    -> select park_name, min(line_price),max(line_price),avg(line_price)
    ->   from sales join sales_line using (transaction_no)
    ->  join themepark using (park_code) group by (park_code);
Query OK, 0 rows affected (0.04 sec)

mysql> select * from ticket_sales;
+--------------+-----------------+-----------------+-----------------+
| park_name    | min(line_price) | max(line_price) | avg(line_price) |
+--------------+-----------------+-----------------+-----------------+
| FairyLand    |           14.99 |          139.96 |       50.232500 |
| PleasureLand |           21.98 |          168.40 |       60.785714 |
| GoldTown     |           12.12 |          114.68 |       47.637778 |
+--------------+-----------------+-----------------+-----------------+
3 rows in set (0.00 sec)
```

Exercise:

Using the date specifiers in Table 7.2, modify the query shown in Figure 55 to display the date in the format 'Fri – 18 – 5 – 07'.

SELECT DISTINCT(DATE_FORMAT(SALE_DATE, "%a-%d-%c-%y")) FROM SALES;

```
mysql> SELECT DISTINCT(DATE_FORMAT(SALE_DATE, "%a-%d-%c-%y")) FROM SALES;
+-----------------------------------------+
| (DATE_FORMAT(SALE_DATE, "%a-%d-%c-%y")) |
+-----------------------------------------+
| Fri-18-5-07                             |
+-----------------------------------------+
1 row in set (0.00 sec)
```

Exercise:

Write a query which generates a list of employee user IDs, using the born month, first day of the month they were born and the first six characters of last name in UPPER case

USER ID format (MDName) here M= month, D= first day of month, Name= Employee last name first 6 alphabets.

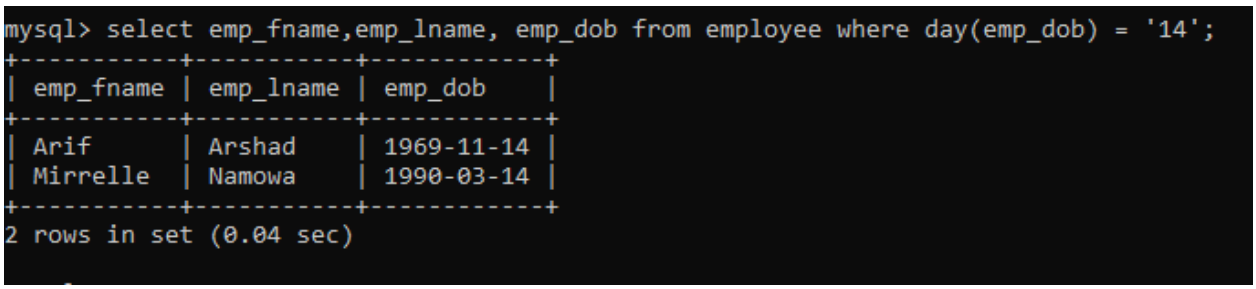
```
select emp_fname,emp_lname, concat

(date_format(emp_dob,'%m01'),

substring( upper(emp_lname),1,6)) as name

 from employee;
```

```
mysql> select emp_fname,emp_lname, concat (date_format(emp_dob,'%m01'),
    -> substring( upper(emp_lname),1,6)) as name
    ->  from employee;
+-----------+------------+------------+
| emp_fname | emp_lname  | name       |
+-----------+------------+------------+
| Emma      | Calderdale | 0601CALDER |
| ibrar     | babar      | 0101BABAR  |
| Arif      | Arshad     | 1101ARSHAD |
| Anne      | Roberts    | 1001ROBERT |
| Enrica    | Denver     | 1101DENVER |
| Mirrelle  | Namowa     | 0301NAMOWA |
| Gemma     | Smith      | 0201SMITH  |
+-----------+------------+------------+
7 rows in set (0.00 sec)
```

Exercise:

Write a query which lists the names and dates of births of all employees born on the 14th day of the month.

```
select emp_fname,emp_lname, emp_dob from employee where
day(emp_dob) = '14';
```



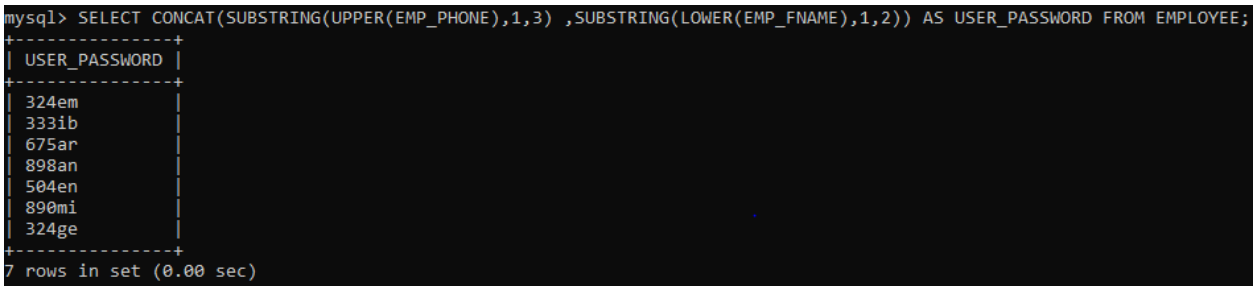
```
mysql> select emp_fname,emp_lname, emp_dob from employee where day(emp_dob) = '14';
+-----------+-----------+------------+
| emp_fname | emp_lname | emp_dob    |
+-----------+-----------+------------+
| Arif      | Arshad    | 1969-11-14 |
| Mirrelle  | Namowa    | 1990-03-14 |
+-----------+-----------+------------+
2 rows in set (0.04 sec)
```

Exercise:

Write a query which generates a list of employee user passwords, using the first three digits of their phone number, and the first two characters of first name in lower case.

Label the column USER_PASSWORD;

SELECT CONCAT(SUBSTRING(UPPER(EMP_PHONE),1,3) ,SUBSTRING(LOWER(EMP_FNAME),1,2)) AS USER_PASSWORD FROM EMPLOYEE;



```
mysql> SELECT CONCAT(SUBSTRING(UPPER(EMP_PHONE),1,3) ,SUBSTRING(LOWER(EMP_FNAME),1,2)) AS USER_PASSWORD FROM EMPLOYEE;
+---------------+
| USER_PASSWORD |
+---------------+
| 324em         |
| 333ib         |
| 675ar         |
| 898an         |
| 504en         |
| 890mi         |
| 324ge         |
+---------------+
7 rows in set (0.00 sec)
```