Database Management System

Unit 6: Retrieving Data using SQL

BHAVIK SARANG | MSC IT | UKA TARSADIA UNIVERSITY

Retrieving Data using SQL

6.2 Summarizing and Grouping Data

6.2.1 Grouping Data:

- Grouping data with Group By.
- Group function

6.2.2 Limiting group data with HAVING

6.3 Numeric Functions

6.4 Character Functions

6.5 Date Functions

6.6 DATETIME Conversion Function

CE: 6.2.1 Grouping Data

6.2.1. Grouping Data with Group By:

- The GROUP BY clause is used to group rows that have the same values.

- Grouping can be done by a column name or with aggregate functions (like SUM, COUNT, etc.), in which the aggregate produces a value for each group.

- Hence, the **GROUP BY** clause creates a data set, containing several sets of records grouped together based on a condition.

- The Group By clause is used in SELECT statements.

6.2.1. Grouping Data with Group By:

- **Syntax:**

SELECT column | expression | group_function(column | expression [alias]),…}

FROM table

[WHERE condition(s)]

[**GROUP BY** {column(s) | expression}]

[ORDER BY {column(s) | expression}] ;

Grouping Data with Group By: (Conti.)

- For Example:

  - **To calculate the number of employees for each designation and average commission for each designation of employees.**

```
Select designation, count(*), avg(comm) from tblEmployee
Group by designation;
```

Grouping Data with Group By: (Conti.)

- For Example:

  - For Table Orders:

    Orders(O_id, Orderdate, Orderprice, Customer)

    Write a query to find the total order of each customer.

  - Query:

    ```
    Select Customer, SUM(Orderprice) from Orders
    GROUP BY Customer;
    ```

Grouping Data with Group By: (Conti.)

- For Example:
- Calculate the total sales for each store in the following table:

Storename	Sales	Date
London	$1600	Jan-01-2014
San Diego	$260	Jan-02-2014
Boston	$300	Jan-03-2014
New Jeshi	$700	Jan-04-2014

- Query:

```
select storename, SUM(sales) from storeinfo GROUP BY
storename;
```

SQL Functions:

- Oracle/MySQL built-in function provides powerful tool for enhancement of a basic query.
- A function takes 0 or more arguments and returns single value.

- There are two types of functions:

1. Single line functions.
2. Multiple line functions/ Grouping functions/ Aggregate functions.

Multiple Line Functions

Aggregate Functions:

1. MAX()
2. MIN()
3. SUM()
4. AVG()
5. COUNT()

Aggregate Functions:

1. MAX():

- It is used to return the maximum value of the expression.
- Syntax:

MAX(<expr>)

- For Example:

SELECT MAX(CurBal) "Maximum Balance" FROM Acct_Master;

Aggregate Functions:

2. MIN():

- It is used to return the minimum value of the expression.

- Syntax:

MIN([<expr>])

- For Example:

```
SELECT MIN(CurBal) "Minimum Balance" FROM Acct_Master;
```

Aggregate Functions:

3. SUM():

- It is used to return the sum of the values of ‘n’.
- Syntax:

```
SUM(<n>)
```

- For Example:

```
SELECT SUM(CurBal) "Total Balance" FROM Acct_Master;
```

Aggregate Functions:

4. AVG():

- It is used to return an average value of ‘n’, ignoring null values in a column.

- Syntax:

  AVG(<n>)

- For Example:

  SELECT AVG(CurBal) "Average Balance" FROM Acct_Master;

Aggregate Functions:

5. COUNT(expr):

- It is used to return the number of rows, where expression in not null.

- Syntax:

  COUNT([distinct] <expr>)

- For Example:

SELECT COUNT(AcctNo) "No. of Accounts" FROM Acct_Master;
SELECT COUNT(distinct city) "No. of Accounts" FROM Acct_Master;

Aggregate Functions:

6. COUNT(*):

- It is used to return the number of rows in the table, including duplicates and those with nulls.

- Syntax:

COUNT(*)

- For Example:

```
SELECT COUNT(*) "No. of Accounts" FROM Acct_Master;
```

International Certification Question

Q1.

For are table:

Sales(Orderid, Orderdate, Orderprice, Orderqty, Customername)

Write a query to retrieve a list of a unique customers from the sales table and at the same time to get the total amount each customer has spent in the store.

Query:

```
select Customername, SUM(Orderprice*Orderqty) from Sales
GROUP BY Customername;
```

International Certification Question

Q2.

Write a query if we wants to returns a list of Department IDs along with the sum of their sales for the date of January 1, 2014.

Query:

select deptid, SUM(Saleamt) from sales where saledate = '01-Jan-2018' GROUP BY deptid;

International Certification Question

Q3.

Write a query by using the COUNT function to return the name of the department. The number of employees(in the associated department) that make over $26,000/year.

Query:

select department, COUNT(*) as "Number of employees" from employees where salary>26000 GROUP BY department;

International Certification Question

Q4.	**The sequence of the columns in a GROUP BY clause has no effect in the ordering of the output.**
A.	**True**
B.	**False**

Ans: B

International Certification Question

Q5.	GROUP BY ALL generates all possible groups - even those that do not meet the query's search criteria.
A.	True
B.	False

Ans: A

Industry Interview Questions

1. What is the use of group by clause?

2. What are aggregate functions or group functions?

Home Work

1. List at least four different aggregate functions. [1 Mark]

2. What is the use of group by clause? [1 Mark]

3. What are the aggregate functions? List at least four aggregate functions supported by MySQL? [2 Marks]

4. Write a note on aggregate function. [5 Marks]

CE: 6.2.2 Limiting group data with HAVING

6.2.2. Limiting Group Data with HAVING:

- The SQL HAVING clause allows us to restrict the data that is sent to the GROUP BY clause.
- Group functions cannot be used in the WHERE clause.
- SQL statement can have both a WHERE clause and an HAVING clause. WHERE filters data before grouping and after grouping.
- HAVING allows a user to perform conditional tests on aggregate values. It is often used in combination with GROUP BY.
- With HAVING, you can include or exclude groups based on the aggregate value for that group.

Limiting Group Data with HAVING: (Conti…)

- Aggregates cannot be used in a WHERE clause, they are used only inside HAVING.

- Syntax:

```
SELECT column_name , aggregate_function (column_name)
FROM    table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function (column_name)  operator value
```

Limiting Group Data with HAVING: (Conti…)

- For Example:
- For finding the average salary for each department that has either more than 20000 employee or starts with a “To”.

```
select dept, avgsal=(avg(salary)) from employee
GROUP BY dept
Having COUNT(Name)>20000 or dept LIKE “To”;
```

International Certification Question

Q1.	The HAVING clause is an optional clause which tells Oracle to group rows based on distinct values that exist for specified columns.
A.	True
B.	False

Ans: B

International Certification Question

Q2.	How many rows of data are returned after executing the following statement?
	SELECT DEPT_ID, SUM(NVL(SALARY,100)) FROM EMP
	GROUP BY DEPT_ID HAVING SUM(SALARY) > 400;
	Assume the EMP table has ten rows and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. The first and second five rows have DEPT_ID values of 10 and 20, respectively.

A.	Two rows
B.	One row
C.	Zero rows
D.	None of the above

Ans: B

International Certification Question

Q2.	**Conti…**

Reason:

- Two groups are created based on their common DEPT_ID values. The group with DEPT_ID values of ten consists of five rows with SALARY values of 100 in each of them.
- Therefore, the SUM(SALARY) function returns 600 for this group, and it satisfies the HAVING SUM(SALARY) > 400 clause.
- The group with DEPT_ID values of 20 has four rows with SALARY values of 100 and one row with a NULL SALARY.
- SUM(SALARY) only returns 400 and this group does not satisfy the HAVING clause.

International Certification Question

Q3.	How many rows of data are returned after executing the following statement? SELECT DEPT_ID, SUM(SALARY) FROM EMP GROUP BY DEPT_ID HAVING SUM(NVL(SALARY,100)) > 400; Assume the EMP table has ten rows and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. The first and second five rows have DEPT_ID values of 10 and 20, respectively.
A.	Two rows
B.	One row
C.	Zero rows
D.	None of the above

Ans: A

International Certification Question

Q3.	**Conti…**

Reason:

- Two groups are created based on their common DEPT_ID values.
- The group with DEPT_ID values of 10 consists of five rows with SALARY values of 100 in each of them.
- Therefore the SUM(NVL(SALARY,100)) function returns 600 for this group and it satisfies the HAVING SUM(SALARY) > 400 clause.
- The group with DEPT_ID values of 20 has four rows with SALARY values of 100 and one row with a null SALARY. SUM(NVL(SALARY,100))
- returns 600 and this group satisfies the HAVING clause.
- Therefore two rows are returned.

Industry Interview Questions

1. What is the use of Having clause?

Home Work

1. What is the difference between WHERE and HAVING Clause?

   [2 Marks]

2. Explain the following clause in SQL with their syntax, options, usage and example.
   a) Group By
   b) Having
   c) Order by
   d) Select
   e) Where

   [6 Marks]

Single Line Functions

Single Line Functions:

- Single Line Functions are:

1. Numeric Functions or Number Functions
2. Character Functions or String Functions
3. Date Functions
4. Conversion Functions

CE: 6.3

Numeric Functions

Numeric functions or Number functions:

1. ABS()
2. POWER()
3. ROUND()
4. SQRT()
5. EXP()
6. EXTRACT()
7. GREATEST()
8. LEAST()
9. MOD()
10. TRUNCATE()
11. FLOOR()
12. CEIL()
13. RAND()

Numeric Functions:

1. ABS():

- It is used to return the absolute value of ‘n’.

- Syntax:

ABS(n);

- For Example:

SELECT ABS(-26) "Absolute“ FROM DUAL;

Numeric Functions:

2. POWER():

- It is used to return **m** raised to the **nth** power. **n** must be an integer, else an error is returned.

- Syntax:

POWER(m,n);

- For Example:

SELECT POWER(6,2) "Raised“ FROM DUAL;

Numeric Functions:

3. ROUND():

- It is used to return **n**, rounded to **m** places to the right of the decimal point. If **m** is omitted, **n** is rounded to **0** places, **m** can be negative to round off digits to the left of the decimal point. **m** must be an integer.

- Syntax:

```
ROUND(n, [m]);
```

- For Example:

```
SELECT ROUND(16.19, 1) "Round" FROM DUAL;
```

Numeric Functions:

4. SQRT():

- It is used to return square root of n.
- Syntax:

```
SQRT(n);
```

- For Example:

```
SELECT SQRT(26) "Square Root" FROM DUAL;
```

Numeric Functions:

5. EXP():

- It is used to return e raised to the nth power, where e = 2.71828183.

- Syntax:

```
EXP(n);
```

- For Example:

```
SELECT EXP(9) "Exponent" FROM DUAL;
```

Numeric Functions:

6. EXTRACT():

- It is used to returns a value extracted from a date or an interval value.

- A DATE can be used only to extract YEAR, MONTH and DAY, while a timestamp with a time zone datatype can be used only to extract TIMEZONE_HOUR and TIMEZONE_MINUTE.

- Syntax:

  Extract({year | month | day | hour | minute | second | timezone_hour | timezone_minute | timezone_region | timezone_abbr} **from** {date_value | interval_value})

Numeric Functions:

6. EXTRACT(): (Conti…)

- For Example:

select **EXTRACT(**year **from** date’2018-07-18’**)** ‘Year’, **EXTRACT(**month **from** now()**)** ‘Month’ FROM DUAL;

Numeric Functions:

7. GREATEST():

- It is used to return the greatest value in a list of expressions.

- Syntax:

  GREATEST(expr1, expr2, ………, expr_n)

  where,

  - expr1, expr2, …...., expr_n are expressions that are evaluated by the greatest function.

- For Example:

SELECT GREATEST(4, 6, 17) "Greatest Number" FROM DUAL;

Numeric Functions:

8. LEAST():

- It is used to return the least value in a list of expressions.

- Syntax:

  LEAST(expr1, expr2, ………, expr_n)

  where,

  - expr1, expr2, …...., expr_n are expressions that are evaluated by the least function.

- For Example:

  SELECT LEAST(4, 6, 17) "Least Number" FROM DUAL;

Numeric Functions:

9. MOD():

- It is used to return the remainder of the first number divided by the second number, both passed as parameters to MOD.

- If the second number is zero, then the result will be same as the first number in **Oracle** and in **MySQL** it will be NULL.

- Syntax:

MOD(m, n)

- For Example:

SELECT MOD(20, 10) “Mod1”, MOD(0, 10) FROM DUAL;

Numeric Functions:

10. TRUNCATE() / TRUNC():

- It is used to returns a number truncated to a certain number of decimal places. The decimal place value is must be an integer.

- If this parameter is omitted, the TRUNCATE function will truncate the number to 0 decimal places.

- Syntax:

TRUNCATE(number, [decimal_places])

- For Example:

SELECT TRUNCATE(126.816, 1) "Trunc1", TRUNCATE(126.816, -1) "Trunc2" FROM DUAL;

Numeric Functions:

11. FLOOR():

- It is used to return a largest integer value that is equal to less than a number.
- Syntax:

FLOOR(n);

- For Example:

```
SELECT FLOOR(24.8) "Flr1", FLOOR(13.16) "Flr2" FROM
DUAL;
```

Numeric Functions:

12. CEIL():

- It is used to return the smallest integer value that is greater than or equal to a number.

- Syntax:

```
CEIL(n);
```

- For Example:

```
SELECT CEIL(24.8) "Ceil1", CEIL(13.16) "Ceil2" FROM DUAL;
```

Numeric Functions:

13. RAND():

- This function returns a random floating-point value in the range from 0 to 1.0. if an integer argument n is specified, it is used as the seed value for producing a repeatable sequence.

- Syntax:

RAND(n)

- For Example:

SELECT RAND() “Rand1”, RAND(30) “RAND2” FROM DUAL;

Numeric Functions:

Several other numberic functions available in Oracle are:

- ACOS(), ASIN(), ATAN(), ATAN2()
- COS(), COSH(), SIN(), SINH(), TAN(), TANH()
- COVAR_POP(), COVAR_SAMP(), VAR_POP(), VAR_SAMP()
- CORR(), SIGN()

Several other numberic functions available in MySQL are:

- PI(), RADIANS(), DEGREES()
- ACOS(), ASIN(), ATAN(), ATAN2()
- COS(), SIN(), TAN(), LOG()
- SIGN()
- LN(), EXP(), COT(), CRC32()

International Certification Question

Q1.	**The second parameter in the ROUND function specifies the number of digits after the decimal point.**
A.	True
B.	False

Ans: A

International Certification Question

Q2.	**Which function is used to return the smallest integer value that is greater than or equal to a number?**
A.	ABS()
B.	CEIL()
C.	FLOOR()
D.	UPPER()

Ans: B

Industry Interview Questions

1. What is the difference between CEIL() and FLOOR()?

2. What is use of ROUND()?

Home Work

1. Which are numeric functions? List at least four character functions. [1 Mark]
2. What is the difference between CEIL() and FLOOR()?

   [2 Marks]
3. List all Numeric functions. Explain any four numeric function in detail.

   [5 Marks]
4. Describe the following functions with its syntax, usages and example:

   ABS, GREATEST, LEAST, TRUNCATE, RAND

   [5 Marks]

CE: 6.4

Character Functions

Character Functions: (In Oracle)

1. ASCII()
2. CONCAT()
3. LENGTH()
4. LOWER()
5. UPPER()
6. INITCAP()
7. LTRIM()
8. RTRIM()
9. TRIM()
10. LPAD()
11. RPAD()
12. SUBSTR()
13. INSTR()
14. TRANSLATE()
15. REPLACE()

Character Functions:

1. ASCII():

- It is use to return the NUMBER code that represents the specified character.
- If more than one character is entered, the function will return the value for the first character and ignore all of the characters after the first.

- Syntax:

ASCII(<single_character>);

- For Example:

SELECT ASCII(‘a’) “ASCII of a”, ASCII(‘b’) “ASCII of b” FROM DUAL;

Character Function:

2. **CONCAT():**

- The CONCAT function joins two character literals, columns, or expressions to yield one larger character expression.
- Numeric and date literals are implicitly cast as characters, when they occur as parameters to the CONCAT function.
- Numeric or date expressions are evaluated before being converted to strings ready to be concatenated.
- The CONCAT function takes two parameters.

Character Function:

2. CONCAT(): (Conti…)

- Syntax:

CONCAT(s1, s2)

  - where s1 and s2 represent string literals, character column values, or expressions resulting in character values.

- For Example:

```
SELECT CONCAT(1+2.14,' APPROXIMATES PI') FROM DUAL;
```

Character Function:

2. CONCAT(): (Conti…)

- CONCAT() can also join three terms to return one character string.
- Since CONCAT() takes only two parameters, it is only possible to join two terms with one instance of this function.
- The solution is to nest the CONCAT function within another CONCAT function.
- For Example:

  select concat('Hiii ! ', concat('Hello',' How are you?')) from dual;

Character Function:

3. LENGTH():

- It is use to return the length of a word.

- Syntax:

```
LENGTH(word)
```

- For Example:

```
select length(mscit');
select length(bmiit ') 'Length' from dual;
```

Character Function:

4. LOWER():

- It is use to return all the character letters in lowercase.

- Syntax:

LOWER(char)

- For Example:

select lower(name) ‘First Name’ from student;

Character Function:

5. UPPER():

- It is use to return all the character letters in upper case.

- Syntax:

UPPER(char)

- For Example:

select upper(name) ‘First Name’ from student;

Character Function:

6. INITCAP():

- It is use to return a string with the first letter of each word in upper case.

- Syntax:

INITCAP(char)

- For Example:

```
SELECT INITCAP('mscit') "Title Case" FROM DUAL;
```

Character Function:

7. LTRIM():

- Returns the string *str* with leading space characters removed.
- Syntax:

  LTRIM(*str*)

- For Example:

  SELECT LTRIM('   TARSADIA') 'Left Trim of Tarsadia' FROM DUAL;

Character Function:

8. RTRIM():

- Returns the string *str* with trailing space characters removed.

- Syntax:

RTRIM(*str*)

- For Example:

```
SELECT RTRIM('TARSADIA          ') "Right Trim of Tarsadia"
FROM DUAL;
```

Character Function:

9. TRIM():

- It is use to remove all specified characters either form the beginning of the ending of a string.

- Syntax:

TRIM([leading | trailing | both [<trim_character> from]]<string1>)

  - *leading* – remove trim_string from the front of string1.
  - *trailing* – remove trim_string from the end of the srting1.
  - *Both* – remove trim_string from the front and end of string1.

- If none of the above option is chosen, the TRIM function will remove trim_string from both the front and end of sting1.

Character Function:

9. TRIM(): (Conti…)

- *trim_character* is the character that will be removed from string1.
- If this parameter is omitted, the trim function will remove all *leading* and *trailing* spaces from string1.
- string1 is the string to trim.
- For Example:

SELECT TRIM(' Bhavik   ‘) FROM DUAL;

Character Function:

9. TRIM(): (Conti…)

- For Example:
  - SELECT TRIM(LEADING 'x' FROM 'xxxBhavikxxx');
  - SELECT TRIM(BOTH 'x' FROM 'xxxBhavikxxx');
  - SELECT TRIM(TRAILING 'xyz' FROM 'Bhavikxxyz');

Character Function:

10. LPAD():

- It is use to return char1, left-padded to length n with the sequence of characters specified in char2.
- If char2 is not specified oracle uses blanks by default.

- Syntax:

```
LPAD (char1, n , [char2])
```

- For Example:

```
SELECT LPAD('Page 1',10,'*') "Lpad" FROM DUAL;
```

Character Function:

11. RPAD():

- It is use to return char1, right-padded to length n with the characters specified in char2.
- If char2 is not specified oracle uses blanks by default.

- Syntax:

```
RPAD(char1, n [,char2])
```

- For Example:

```
select RPAD(name,10,'x') "RPAD Example" from student;
```

Character Function:

12. SUBSTR():

- It is use to return a portion of characters, beginning at character **m**, and going upto character **n**.

- If **n** is omitted, the result returned is upto the last character in the string. The first position of char is 1.

- Syntax:

  SUBSTR(<string>, <start_position>, [<length>])

  - *string* is the source string.
  - *start_position* is the position for extraction. The first position in the string is always 1.
  - *length* is the number of characters to extract.

Character Function:

12. SUBSTR():

- For Example:

```
SELECT SUBSTR('Database', 5, 3) FROM DUAL;
```

Character Function:

13. INSTR():

- It is use to return the location of a substring in a string.

- Syntax:

  INSTR(<string1>, [<nth_appearance>]);

  - *string1* is the string to search.
  - *string2* is the substring to search for in string1.
  - *start_position* is the position in string1, where search will start.
- If omitted, it defaults to 1.
- The first position in the string is 1.

Character Function:

13. INSTR(): (Conti…)

- If the start_position is negative, the function counts back start_postion number of characters from the end of string1 and then searches towards the beginning of string1.

- The *nth_appearance* is the nth appearance of string2.
- If omitted, it defaults to 1.

- For Example:

```
SELECT INSTR('SCT on the net', 't') "Instr1" FROM DUAL;
```

Character Function:

14. TRANSLATE():

- It is use to replace a sequence of characters in a string with another set of characters.
- It replaces a single character at a time.

- Syntax:

  Translate(<string1>, <string_to_replace>, <replacement_string>)

  - *string1* is the string to replace a sequence of characters with another set of characters.
  - *string_to_replace* is the string that will be searched for in string1.
  - All characters in the *string_to_replace* will be replaced with the corresponding character in the *replacement_string*.

Character Function:

14. TRANSLATE(): (Conti…)

- For Example:

```
SELECT TRANSLATE(‘tmt bscet’, ‘tmbe’, ‘inmi’) FROM DUAL;
```

Character Function:

15. REPLACE():

- The REPLACE function replaces all occurrences of a search item in a source string with a replacement term and returns the modified source string.

- If the length of the replacement term is different from that of the search item, then the lengths of the returned and source strings will be different.

- If the search string is not found, the source string is returned unchanged.

Character Function:

15. REPLACE(): (Conti….)

- The REPLACE function takes three parameters.
- Syntax:

  REPLACE(source string, search item, <replacement term>);

- If the replacement term parameter is omitted, each occurrence of the search item is removed from the source string.
- In other words, the search item is replaced by an empty string.

Character Function:

15. REPLACE(): (Conti….)

- For Example:

```
SELECT REPLACE(10000-3,'9','86');

SELECT REPLACE(SYSDATE, 'DEC','NOV');

SELECT REPLACE('1#3#6#7#9#','#','->');

SELECT REPLACE('1#3#6#7#9#','#');
```

Character Function: (In MySQL)

1. ASCII()
2. CONCAT()
3. CONCAT_WS()
4. LENGTH()
5. LOWER()/ LCASE()
6. UPPER()/ UCASE()
7. LEFT()
8. MID()
9. RIGHT()
10. LTRIM()
11. RTRIM()
12. TRIM()
13. LPAD()
14. RPAD()
15. SUBSTR()/ SUBSTRING()
16. INSTR()
17. REPLACE()
18. LOCATE()
19. BIT_LENGTH()
20. CHAR_LENGTH()
21. SOUNDEX()
22. REVERSE()
23. REPEAT()

Character Function:

7. LEFT():

- It is use to return the left starting characters of the string.

- Syntax:

  LEFT (<char>, <m> [,<n>])

- For Example:

  SELECT LEFT('Tarsadia', 5) 'Left';

Character Function:

8. MID():

- It is use to return the substring starting from the specified position of the string.
- Syntax:

  MID (<char>, <m> [,<n>])
- For Example:

  SELECT MID('Tarsadia', 5,2) 'Middle';

Character Function:

9. RIGHT():

- It is use to return specified rightmost number of characters.
- Syntax:

  RIGHT (<char>, <m> [,<n>])
- For Example:

  SELECT RIGHT('Tarsadia', 5) 'Right';

Character Function:

18. LOCATE():

- It is use to return the position of the first occurrence of substring.

- Syntax:

  LOCATE (<char1>, string [, startat])

- For Example:

  SELECT fname, LOCATE('a', fname), LOCATE('a', fname, 3) from Employee;

Character Function:

19. BIT_LENGTH():

- It is use to return the length of the string **str** in bits..

- Syntax:

  BIT_LENGTH (<str>)

- For Example:

  SELECT BIT_LENGTH('Bhavik') from dual;

Character Function:

20. CHAR_LENGTH():

- It is use to return the length of the string **str** in bits..
- Syntax:

```
CHAR_LENGTH (<str>)
```

- For Example:

```
SELECT CHAR_LENGTH('Bhavik') from dual;
```

Character Function:

21. SOUNDEX():

- A soundex is a phonetic algorithm for indexing names after English pronunciation of sound.
- All non-alphabetic characters in a string are ignored.
- All international alphabetic characters outside the A-Z range are treated as vowels.
- We can use SUBSTRING() on the result to get a standard soundex string.

Character Function:

21. SOUNDEX(): (Conti…)

- Syntax:

SOUNDEX(<string>)

- For Example:

SELECT SOUNDEX('Bhavik') from dual;

SELECT SOUNDEX(fname) from tblStudent;

SELECT fname from tblStudent where SOUNDEX(fname) = SOUNDEX(‘Bhavik’);

SELECT * from tblStudent where SOUNDEX(fname) LIKE ‘%a%’;

Character Function:

22. REVERSE():

- It is use to return the reverse of a string specified as an argument.

- Syntax:

  REVERSE(<string>)

- For Example:

```
SELECT REVERSE('Bhavik') from dual;
SELECT REVERSE(fname) from tblStudent;
```

Character Function:

23. REPEAT():

- It is use to repeat a string for a specified number of times.
- Both the *string* and the *number of times that string to be repeated* are specified as arguments.
- The function returns NULL, if any of the arguments is NULL.
- Syntax:

REPEAT(<string>, <no. of times>)

Character Function:

23. REPEAT(): (Conti…)

- For Example:

```
SELECT REVERSE(fname, 2) from tblStudent;
```

Character Functions:

Several other character functions available in Oracle are:

- COMPOSE(), DECOMPOSE(), VSIZE()

Several other character functions available in MySQL are:

- BIN(), QUOTE(), SPACE(), OCT(), ORD()
- COMPRESS(), UNCOMPRESS(), UNCOMPRESS_LENGTH()
- CONV(), ELT(), EXPORT_SET(), FIELD(), FIND_IN_SET(), MAKE_SET()
- HEX(), UNHEX()
- LOAD_FILE(), OCTET_LENGTH(), POSITION(), REGEXP, STRCMP(), SUBSTRING_INDEX(), CHAR(), CHARACTER_LENGTH(), FORMAT(), INSERT()

International Certification Question

Q1.	Which function joins two character literals, columns, or expressions to yield one larger character expression?
A.	CONCAT()
B.	ANSII()
C.	CHR()
D.	Non of These

Ans: A

International Certification Question

Q2.	Which statements regarding single-row functions are true?
A.	They may return more than one result.
B.	They execute once for each record processed.
C.	They may have zero or more input parameters.
D.	They must have at least one mandatory parameter.

Ans: B and C

International Certification Question

Q3.	The LOWER function returns a char, with all letters in lowercase.
A.	True
B.	False

Ans: A

International Certification Question

Q4.	The UPPER function returns a string with the first letter of each word in upper case.
A.	True
B.	False

Ans: B

International Certification Question

Q5.	**The LENGTH function returns the length of a word.**
A.	**True**
B.	**False**

Ans: A

International Certification Question

Q6.	The LTRIM returns char, with final characters removed after the last character not in the set. ‘set’ is optional, it defaults to spaces.
A.	True
B.	False

Ans: B

International Certification Question

Q7.	LPAD returns the string passed as a parameter after left padding it to a specified length.
A.	True
B.	False

Ans: A

Industry Interview Questions

1. What is the use SQL function? List which are SQL function.
2. What is the difference between RPAD() and RTRIM() function?
3. What is the difference between INITCAP() and UPPER()?

Home Work

1. What is the use of SQL function? List which are SQL functions.

   [2 Marks]

2. Which are character functions? List at least four character functions. [2 Marks]

3. Describe following character functions with its syntax, usages and example:

   LENGTH, INSTR, LPAD, RPAD, TRIM

   [5 Marks]

CE: 6.5
Date Functions

Date Functions:

1. NOW()
2. SYSDATE()
3. CURDATE()
4. CURTIME()
5. DATE_ADD / DATE_SUB
6. ADDDATE / SUBDATE
7. LAST_DAY()
8. PERIOD_DIFF()

Date Functions:

1. NOW():

- Now() is a function that requires no arguments and returns the current date and time.

- For Example:

```
SELECT NOW();
```

Date Functions:

2. SYSDATE():

- SYSDATE is used to return the date of the local database.
- The SYSDATE can be used just as any other column name.
- It takes no arguments.

- For Example:

```
SELECT SYSDATE();
```

Date Functions:

3. CURDATE() / CURRENT_DATE():

- It returns the current date as a value in YYYY-MM-DD or YYYYMMDD format.
- The format depends on whether the function is used with a date expressed as a string or numeric value.
- It takes no arguments.

- For Example:

```
SELECT CURRENT_DATE();
SELECT CURRENT_DATE() + 0;
```

Date Functions:

4. CURTIME():

- It is used to return the system current time as a value in HH:MM:SS or HHMMSS format, depending on whether used with time expressed as a string or numeric value.
- It takes no arguments.
- For Example:

```
SELECT CURTIME(), CURTIME() + 0 "Number Format";
```

Date Functions:

5. DATE_ADD / DATE_SUB:

- It is used to return a date after adding / subtracting the value specified as a parameter to the function.

- Syntax:

```
DATE_ADD(<date>, INTERVAL <value> <type>)
DATE_SUB(<date>, INTERVAL <value> <type>)
```

- For Example:

```
SELECT CURRENT_DATE, DATE_ADD(CURRENT_DATE,
INTERVAL 2 MONTH);
SELECT DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR);
```

Date Functions:

6. ADDDATE / SUBDATE:

- It is also used to return a date after adding / subtracting the value specified as a parameter to the function.

- Syntax:

  ADDDATE (<date>, INTERVAL <value> <type>)

  SUBDATE (<date>, INTERVAL <value> <type>)

- Syntax:

  ADDDATE (<expression>, <days>)

  SUBDATE (<expression>, <days>)

- Where, **Expression** is a DATE or DATETIME expression and **Days** is the number of days to be added to **Expression**.

Date Functions:

6. ADDDATE / SUBDATE (Conti…):

- For Example:

```
SELECT CURRENT_DATE, ADDDATE(CURRENT_DATE,65);
```

Date Functions:

7. LAST_DAY():

- It is used to return the last date of the month passed as a parameter to the function.

- Syntax:

LAST_DAY(<Date>)

- For Example:

SELECT CURRENT_DATE, LAST_DAY(CURRENT_DATE) "Last Date";

Date Functions:

8. PERIOD_DIFF():

- It is used to return the number in months between two given date values.

- Syntax:

  PERIOD_DIFF(<Period1>, <Period2>)

  - Where, Period1 and Period2 should be either in YYYYMM or YYMM format only.

- For Example:

```
SELECT PERIOD_DIFF('201801','201701') "Months";
```

CE: 6.6
DATETIME Conversion Functions

DATETIME Conversion Functions:

- MySQL has following DATETIME conversion functions:
  1. DATE_FORMAT()
  2. TIME_FORMAT()

1. DATE_FORMAT():

- This function is used to formats the date value according to the format string specified.

- Syntax:

DATE_FORMAT(<date>, <format>)

DATETIME Conversion Functions:

1. DATE_FORMAT(): (Conti…)

- For Example:

SELECT DATE_FORMAT(‘2016-08-09 12:23:00’, ‘%D %W %M %Y’);

Options to format DATES and TIME:

Format	Description
%a	Abbreviated weekday name (Sun-Sat)
%b	Abbreviated month name (Jan-Dec)
%c	Month, numeric (0-12)
%D	Day of month with English suffix (0th, 1st, 2nd, 3rd, ….)
%d	Day of month, numeric (00-31)
%e	Day of month, numeric (0-31)
%f	Microseconds (000000-999999)
%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)

Options to format DATES and TIME: (Conti…)

Format	Description
%M	Month name (January-December)
%m	Month, numeric (00-12)
%p	AM or PM
%r	Time, 12-hour (hh:mm:ss followed by 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 (Sunday-Saturday)
%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

Options to format DATES and TIME: (Conti…)

Format	Description
%x	Year for the week where Monday is the first day of week, four digits, used with %v
%Y	Year, numeric, four digits
%y	Year, numeric, two digits

DATETIME Conversion Functions:

1. DATE_FORMAT(): (Conti…)

- For Examples:

```
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')
```

- The result would look something like this:

```
Nov 04 2014 11:45 PM
11-04-2014
04 Nov 14
04 Nov 2014 11:45:34:243
```

DATETIME Conversion Functions:

2. TIME_FORMAT():

- This function is used to formats the date value according to the format string specified.

- Syntax:

  TIME_FORMAT(<time>, <format>)

- For Example:

  SELECT TIME_FORMAT(’14:56:15’, ‘%H %k %h %I %i’);

Home Work

1. What is the use of ADDDATE? [1 Mark]
2. Differentiate NOW() and CURDATE(). [2 Marks]
3. Describe the following date functions with its syntax, usages and example.

   CURDATE, CURTIME, DATE_ADD, SUBADD, LAST_DAY.

   [5 Marks]

4. Which are Conversion Function? Explain each briefly with example.

   [5 Marks]

Thank You