MIS 301 Relational Database Management System

DATABASE MANAGEMENT SYSTEM

Structured Query Language(SQL)-1

LECTURE 8&9

Structured Query Language(SQL)

- SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.
- SQL is the standard language for Relational Database System and operates on **relational databases**.
- SQL is based on relational algebra and relational calculus.
- SQL commands are used for defining data structures and manipulating data stored in them i.e. adding data, changing data, deleting data, seeing and removing data.
- SQL commands are broadly classified as DDL, DML, DCL and TCL.

STRUCTURED QUERY LANGUAGE(SQL) DATATYPES

- ***char(size)***→ Fixed length string 0 to 255 characters long. Default size is 1. It can contain digits, alphabets and special characters. No arithmetic is possible.
- ***varchar(size)***→Variable length string 0 to 65535 characters long storing digits, alphabets and special characters. Default size is 1.
- ***binary(size)***→Similar to char but stores binary byte strings. Default size is 1. char strings are directly readable by us but must be encoded to binary byte strings to be comprehended by the computer. Pictures, music, text, etc are all encoded and stored as byte strings.

STRUCTURED QUERY LANGUAGE(SQL)
DATATYPES

- ***bit(size)***→stores bits, number of bits per value is the size which ranges from 1 to 64 (i.e. 0 or 1 for each bit) . Default value is 1.
- ***integer(size)/int(size)***→stores integer values in the range -2147483648 to 2147483647 . If the integer is unsigned, the range is 0 to 4294967295.
- ***float(size,dad)***→stores real numbers i.e. includes decimal fractions within 32 bits. Size refers to total number of digits and dad refers to number of **d**igits **a**fter the **d**ecimal point.
- ***double(size,dad)***→stores real numbers i.e. includes decimal fractions within 64 bits. Size refers to total number of digits and dad refers to number of **d**igits **a**fter the **d**ecimal point.
- ***decimal(size,dad)/dec(size,dad)***→stores real numbers i.e. includes decimal fractions within 128 bits. Size refers to total number of digits and dad refers to number of **d**igits **a**fter the **d**ecimal point.
- ***bool***→It is used to specify Boolean values true and false. Zero is considered as false, and nonzero values are considered as true.

STRUCTURED QUERY LANGUAGE(SQL)
DATATYPES

- ***number(size,dad)*** →stores real numbers i.e. includes decimal fractions within 38 digits. Size refers to total number of digits and dad refers to number of **d**igits **a**fter the **d**ecimal point which ranges between 84 and 127. If dad is 0, the number is an integer.
- ***Date*** →It is used to store a valid date(DD-MMM-YY) format with a fixed length. Its range varies from January 1, 4712 BC to December 31, 9999 AD.

STRUCTURED QUERY LANGUAGE(SQL)
OPERATORS-ARITHMETIC

- **+ → *addition(example a+b)***
- **- → *subtraction(example a-b)***
- **** → multiplication(example a*b)***
- ***/ → division(example a/b)***
- ***% → remainder after integer division***

Let a=20, b=6

Therefore
a+b=26
a-b=14
a*b=120
a/b=2.33
a%b=2

STRUCTURED QUERY LANGUAGE(SQL)
OPERATORS-COMPARISON/RELATIONAL

- ❖***These operators always return*** *true* ***or*** *false.*
- ❖***They compare their operands.***

- ***=   → equal to (example a=b)***
- ***!= or <>→ not equal to(example a<>b)***
- ***<   → less than(example a<b)***
- ***<= → less than or equal to(example a<=b)***
- ***>   → greater than(example a>b)***
- ***>= →greater than or equal to(example a/b)***

Let a=20, b=6

Therefore
a=b gives false
a<>b gives true
a!=b gives true
a<b gives false
a<=b gives false
a>b gives true
a>=b gives true

STRUCTURED QUERY LANGUAGE(SQL)

OPERATORS-LOGICAL

- ***These operators also return** true **or** false.*
- ***Their operands are logical operations or values***

- ***and → returns true if both operands are true(example a=b and a<c)***
- ***or → returns true if atleast one of its operands is true(example a=b or a<c)***
- ***not → returns true if its operand is false and false if it is true(example not a<b)***

Let a=20, b=6, c=20

Therefore
a=b and a=c gives false
a=b or a=c gives true
not a=c gives false
not a=b gives true

STRUCTURED QUERY LANGUAGE(SQL)

OPERATORS-LOGICAL

- *between and* → ***returns true if given value falls within given limits, including limits(example a between 10 and 20)***
- *in* → ***returns true if given value belongs to a given set{example a in (10,15,20,25,30) or city in ('Kolkata', 'Durgapur', 'Patna')}***
- *like* → ***returns true if given value matches a pattern satisfying wildcards. Exact match is not looked for(example city like '%pur')***
- *all* →**The ALL operator returns TRUE if and only if all of the subqueries' values meet the condition. The ALL must be preceded by comparison operators and evaluates to true if all of the subqueries values meet the condition. It is used with SELECT, WHERE, HAVING statement. {example select roll_no from student where marks> all (select marks from batch_last);}**

STRUCTURED QUERY LANGUAGE(SQL)
OPERATORS-LOGICAL

- *any* → **ANY compares a value to each value in a list or results from a query and evaluates to true if the result of an inner query contains at least one row. ANY returns true if any of the subquery values meet the condition. It must be preceded by comparison operators.{Example select roll_no, name from student where city =any(select city from batch_last;}**
- *is null* → ***returns true if the given attribute value is null i.e. empty.{*****Example** ***select * from student where marks is null;}***
- *exists* → **The EXISTS operator is used to test for the existence of any record in a subquery. The EXISTS operator returns true if the subquery returns one or more records.{Example select name, stream from student where exists (select city from metros where student.city=metros.city); }**

- TILL WE MEET AGAIN IN THE NEXT CLASS……….