Lab 4

SQL Operators

There are four types of operators in SQL:

- **Arithmetic operators: +, *, /, %**
- **Comparison operators: =, ! =, <>, >, <, >=, <=, !<, !>**
- **Logical Operators: AND, OR, NOT, BETWEEN, IN, LIKE, IS NULL**

- **E.g. Select * from game where location is null;**

Distinct keyword - The SQL DISTINCT keyword is used in conjunction with SELECT statement to eliminate all the duplicate records and fetching only unique records.

SELECT DISTINCT SALARY FROM CUSTOMERS ORDER BY SALARY;

Concatenate operator - ||

Select id||'_'||name as employee_name from employee;

PostgreSQL built-in functions

Aggregate functions – count, max, min, sum, avg

Numeric functions – X is the name of your numeric type attribute

Ceiling (X) return the smallest integer value that is not smaller than X,

Floor (X) returns the largest integer value that is not greater than X,

Power (X, 2) return the value of X raised to the power of 2,

Round (X) returns X rounded to the nearest integer

Round (X, 2) returns X rounded to the nearest 2 decimal places

String functions – X is the name of your varchar type attribute

lower (X), length (X), upper(X), reverse (X), Lpad (X, length,'??'), Rpad (X, length,'??'), SUBSTRING (X, pos), SUBSTRING (X FROM pos), SUBSTRING (X, pos, len),
SUBSTRING (X FROM pos FOR len),
Ltrim(X, string to be trimmed)
Rtrim(X, string to be trimmed)
Trim (X, string to be trimmed)

Date functions –

Age (X): select age(timestamp '1957-06-13');

Age (X, Y): SELECT AGE(timestamp '2001-04-10', timestamp '1957-06-13');

```
SELECT now();
```

The valid field names are: century, day, decade, dow, doy, hour, isodow, isoyear, microseconds, millennium, milliseconds, minute, month, quarter, second, timezone, timezone_hour, timezone_minute, week, year.

```
SELECT date_part('day', TIMESTAMP '2001-02-16 20:38:40');
Select to_char (date_of_join,’DD/Month/YYYY’) from employee;
```

```
select extract(year from date_of_join) as output from employee
```

YYYY	4-digit year
Y,YYY	4-digit year, with comma
YYY YY Y	Last 3, 2, or 1 digit(s) of year
Q	Quarter of year (1, 2, 3, 4; JAN-MAR = 1).
MM	Month (01-12; JAN = 01).
MON	Abbreviated name of month in all uppercase

Mon	**Abbreviated name of month capitalized**
mon	**Abbreviated name of month in all lowercase**
MONTH	**Name of month in all uppercase, padded with blanks to length of 9 Characters**
Month	**Name of month capitalized, padded with blanks to length of 9 Characters**
month	**Name of month in all lowercase, padded with blanks to length of 9 Characters**
RM	**Month in uppercase Roman numerals**
rm	**Month in lowercase Roman numerals**
WW	**Week of year (1-53) where week 1 starts on the first day of the year**
W	**Week of month (1-5) where week 1 starts on the first day of the month**
IW	**Week of year (01-53) based on the ISO standard**
DAY	**Name of day in all uppercase, padded with blanks to length of 9 Characters**
Day	**Name of day capitalized, padded with blanks to length of 9 characters**

day	**Name of day in all lowercase, padded with blanks to length of 9 Characters**
DY	**Abbreviated name of day in all uppercase**
Dy	**Abbreviated name of day capitalized**
dy	**Abbreviated name of day in all lowercase**
DDD	**Day of year (1-366)**
IDDD	**Day of year based on ISO year**
DD	**Day of month (01-31)**

The `to_date()` function converts a string literal to a date value. The following illustrates the syntax of the `to_date()` function:

Consider the employee table:
Employee (employee_id varchar(15), ename varchar(20), date_of_birth date, salary numeric(9,2))

1. Find the ceiling value for the salary of employees.
2. Find the floor value for the salary of employees.
3. Round off the salary of employees to the nearest 2 places.
4. Represent the value of salary raised to the power of 2.
5. Represent the name of employees in lower case.
6. Display the name of the employees along with the string length.
7. Pad the extra space of name of employees with '*' on the left.
8. Pad the extra space of name of employees with '*' on the right.
9. Right trim spaces (if any) from the name of employees.
10. Left trim spaces (if any) from the name of employees.
11. Pick 3 characters from the second position of the name of employees.
12. Use the to_char function to format the date of birth field of employees.
13. SELECT to_date('20170103','YYYYMMDD');
14. Suppose you want to convert the string 2017 Feb 10 to a date value, you can apply the pattern YYYY Mon DD as follows:

```
SELECT to_date('2017 Feb 20','YYYY Mon DD');
```

15. Find the employees who celebrate their birthday in January.