

Oracle-SQL

Contents

- Introduction
- Commands
- Constraints
- Operators
- Clauses
- Functions
- Case/Decode
- Joins
- Sub Queries
- Pseudo Columns
- Who Columns
- List Aggregate
- With Clause
- Views
- Sequences
- SQL * Loader

Introduction

- What is a Data?
- What is Database?
- Why do we need Database?
- What will happen if we don't store data?
- What is SQL?
- What is NOSQL?
- Other DB Platforms?

SQL Commands

- **DDL:** Data Definition language
- **DML:** Data Manipulation Language
- **DRL/DQL:** Data Retrieve/Query Language
- **TCL:** Transaction Control Language
- **DCL:** Data Control Language

DDL

- Create
- Alter (Add, Drop, Rename, Modify)
- Rename
- Drop
- Truncate

DML

- Insert
- Update
- Delete

DRL/DQL

➤ Select

TCL

- Commit
- Rollback
- Save point

DCL

- Grant
- Revoke

Datatypes

- **Numeric:** NUMBER
- **Character:** VARCHAR, VARCHAR2, CHAR, NCHAR, NVARCHAR, etc...
- **Date:** TIMESTAMP, DATE

Constraints

- Primary Key
- Unique
- Not Null
- Check
- Foreign Key

SQL Operators

- **Arithmetic Operators:** +, -, *, /
- **Logical Operators:** AND, OR, NOT
- **Comparison:** =, != or <>, <, >, >=, <=, BETWEEN, IN, NOT IN, IS NULL, IS NOT NULL
- **Concatenation:** Concat or Pipe (||)
- **Set Operators:** UNION, UNION ALL, INTERSECT, MINUS
 - MINIMUM TWO QUERIES ARE REQUIRED
 - COLUMN COUNT SHOULD MATCH IN ALL THE QUERIES
 - COLUMN TYPE SHOULD MATCH IN ALL THE QUERIES
 - UNION: RETURNS THE NON-DUPLICATE VALUES
 - UNION ALL: RETURNS THE COMBINED RESULTS FROM TWO TABLES
 - INTERSECT: RETURNS COMMON RECORDS FROM TWO TABLES
 - MINUS: RETURNS WHICH ARE PRESENT IN FIRST TABLE AND NOT PRESENT IN SECOND TABLE
- **Like Operator**
 - % is used to replace multiple characters
 - _(underscore) is used to replace single character

Clauses

- SELECT
- FROM
- WHERE
- GROUP BY
- HAVING
- ORDER BY

Order of Execution

- FROM
- WHERE
- GROUP BY
- HAVING
- SELECT
- ORDER BY

Order By Clause:

- Order by is used to sort the data in Ascending/Descending order
- Default Order is Ascending Order
- Asc or Desc

Single Column:

- Order By Column1 Asc/Desc

MultiColumn:

- Order By Column1 Asc/Desc, Column2 Asc/Desc

With Numbers:

- Order By 1 Asc/Desc, 2 Asc/Desc

Group by Clause:

- Used to group the records

With Aggregate Functions:

Single Column:

Multi Column:

Without Aggregate Functions:

Single Column:

Multi Column:

Functions:

- Numeric Functions
- Aggregate Functions
- Character/String Functions
- Date Functions
- Null/Compare Functions
- Analytic Functions

Numeric Functions

- Ceil
- Floor
- Round
- Sign
- Mod
- TRUNC
- Abs
- sqrt
- Power

Rules:

- Used for numeric types
- Round(number, decimal), TRUNC(number, decimal)
- If decimal is +ve right direction after decimal point
- If decimal is -ve left direction after decimal point

Aggregate Functions

- AVG
- MIN
- MAX
- SUM
- COUNT

Rules:

- Always produces single output
- mostly used with group by clause
- Columns which are used in select clause with aggregate function must be available in group by clause

Character/String functions

- **ASCII:** Gives the ASCII code for char
- **CHR:** Gives the CHR values for ASCII code (vice versa of ASCII)
- **CONCAT:** Used to append 2 strings
- **INITCAP:** Converts all initial letters into capital in a sentence
- **LOWER:** Converts to all characters to lower case
- **UPPER:** Converts to all characters to upper case
- **LPAD:** 3 parameters (input, length, adding)
- **RPAD:** 3 parameters (input, length, adding)
- **LTRIM:** 2 parameters (input, trim characters)
- **RTRIM:** 2 parameters (input, trim characters)
- **TRIM:** 1 parameter (position character from string)
- **SUBSTR:** 3 parameters (input, position, length)
- **INSTR:** 4 parameter (input, substr, position, occurrence)
- **LENGTH:** 1 parameter (string)

Character functions Contd.

- **REPLACE:** 3 parameters (input, old, new) replace string by string
- **TRANSLATE:** 3 parameters (input, old, output) replace set of characters, character by character

NULL functions

- **NVL:** 2 parameters, Returns second value if the first value is null
- **NVL2:** 3 parameters, returns 3rd value if first value is null or else it returns second value
- **COALESCE:** Accepts “n” no of parameters and returns first non-null value
- **NULLIF:** 2 parameters , if two values are equal returns null or else returns first value

Date Functions

- **SYSDATE:** returns operating system of the database server
- **CURRENT_DATE:** returns current date in the session time zone
- **CURRENT_TIMESTAMP:** returns current date and time in the session time zone
- **SYSTIMESTAMP:** returns current date and time in the database server
- **LOCALTIMESTAMP:** returns current timestamp in session without time zone
- **SESSIONTIMEZONE:** Shows the time zone of the area

Date Functions Contd.

- **TO_DATE:** 2 parameters (appropriate date format ,specified date format)
- **ADD_MONTHS:** 2 parameters (date, no. of months) finally adds months to the date
- **MONTHS_BETWEEN:** 2 parameters ,(date1, date2) and returns diff in months
- **LAST_DAY:** 1 parameter, LAST_DAY(date) and returns the last day of the month
- **NEXT_DAY:** 2 parameters, NEXT_DAY(date, 'day') and returns next coming day date
- **LEAST:** N parameters, returns least date from multiple dates
- **GREATEST:** N parameters, returns the greatest date from multiple dates
- **TO_CHAR:** 2 parameters, converts oracle date to character date format
- **EXTRACT:** To extract day, year, month from date
- **NEW_TIME:** 3 parameters, Converts date from one time zone to another time zone
- **COALESCE:** N parameters, Returns first non null value

Date Functions Contd.

ROUND: 2 parameters, 2nd is optional

- ✓ If 2nd parameter is year, it will return to the nearest year
- ✓ If 2nd parameter is month, it will return to 1st date of the nearest month
- ✓ If 2nd parameter is day, it will return nearest SUNDAY
- ✓ If 2nd parameter is quarter, it will return 1st date of the nearest quarter month
- ✓ If 2nd parameter is week, SATURDAY

TRUNC: 2 parameters, 2nd is optional

- ✓ If 2nd parameter is year, returns first date of the year
- ✓ If 2nd parameter is month, returns first date of the months
- ✓ If 2nd parameter is day, returns last Sunday date
- ✓ If 2nd parameter is quarter, returns first date of the quarter month
- ✓ If 2nd parameter is week, returns same day of the past week.

Analytic Functions

- **ROW_NUMBER:** Always return sequence numbers
- **RANK:** If duplicates found assigns the same rank, and skips the sequence number
- **DENSE_RANK:** If duplicates found assigns the same rank, and continues the sequence number
- **LEAD:** 3 parameters, Returns the next value
- **LAG:** 3 parameters, Returns the previous value

Pseudo columns

- **ROWNUM**: It gives the sequential rownum to the rows and used to get mentioned records only
- **ROWID**: It's a 16 digit physical address and used to identify and delete duplicate records

Rules:

- **ROWNUM** is used to limit the records

JOINS

- **EQUI JOINS:** It combines rows of one table associated with one or more rows in another table based on the equality of column values or expressions.
- **NON-EQUI JOINS:**
- **CROSS JOIN:**
- **OUTER JOIN:**
- **LEFT OUTER:**
- **RIGHT OUTER:**
- **FULL OUTER:**
- **INNER JOIN:**
- **SELF JOIN:**
- **NATURAL JOIN:**

Sub Query:

A query with in the another query is called subquery

Types:

1. Non-Correlated Subqueries
 - ✓ Single Row Subquery
 - ✓ Multi Row Subquery
 - ✓ Multi Column Subquery (Inline View)
2. Correlated Subqueries

Rules:

Subquery can be placed or written in following clauses

- ✓ Select Clause - Single/Multi Row Subquery - Single Column Subquery
- ✓ From Clause - Multi Column Subquery
- ✓ Where Clause - Single/Multi Row Subquery - Single Column Subquery

- **CASE**
- **DECODE**
- **LISTAGG**
- **DISTINCT**
- **WITH CLAUSE**

LISTAGG:

Used to get all the row data into a single result separated by delimiter

Syntax:

```
LISTAGG(ENAME, ',') WITHIN GROUP(ORDER BY ENAME) OVER(PARTITION BY DEPT_NO) NAMES
```

WITH CLAUSE

Sub Query Refactoring

CTE Common table expression

1. Used to reuse sub queries
2. Used to reduce the complexity of the query

Syntax:

WITH <Table Name> (Columns) AS (Sub Query) Select Statement.

**WITH <Table Name> (Columns) AS (Sub Query1), (Sub Query2)
Select Statement.**

Who Columns

- WHO columns are used to track the information updated or inserted by the users against the tables. FND_STANDARD package is used for this purpose. FND_STANDARD.SET_WHO Procedure is used to update the WHO columns in a Table when a DML operations (i.e. INSERT, UPDATE) performed.
- Created by
- Creation date
- Last_updated_by
- last_update_date
- last_update_login


```
SELECT * FROM v$version;
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
```