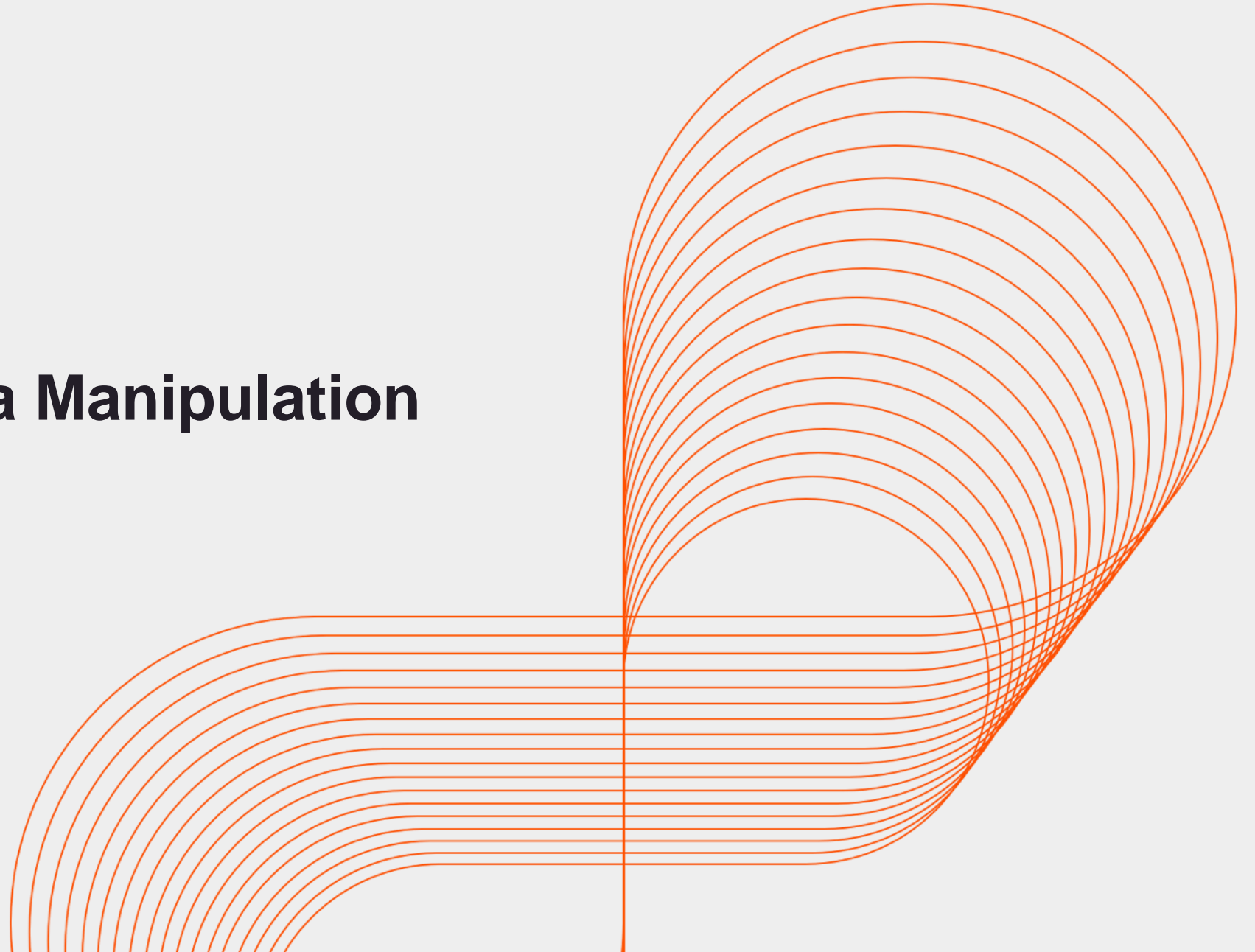




# Nugget 3: Data Manipulation

Persistent University



## Key Learning Points

- 1. Meaning of Data Manipulation Language (DML)**
- 2. Types of DML statements**
- 3. Use of Substitution variables with DML**

## Sample Data

**Table Name : Employee**

| EMPLOYEEID | FIRSTNAME | LASTNAME | EMAIL                   | PHONENUMBER | HIREDATE  | JOBID      | SALARY | COM<br>MITIO<br>NPCT | MANAGERID | DEPARTMENTID |
|------------|-----------|----------|-------------------------|-------------|-----------|------------|--------|----------------------|-----------|--------------|
| 1          | John      | Demn     | JohnD@yah<br>oo.com     | 9898780979  | 1/10/2001 | IT_PROF    | 70000  | 0.5                  | NULL      | 10           |
| 2          | Ken       | Dale     | kendaleD@g<br>mail.com  | 7877787655  | 4/1/2001  | SALES_HEAD | 50000  | NULL                 | NULL      | 10           |
| 3          | James     | Walton   | JW@yahoo.<br>com        | 5787887888  | 1/1/2001  | IT_REP     | 30000  | 0.2                  | 1         | 20           |
| 4          | robin     | sngal    | robin@gmail<br>.com     | 4990988839  | 5/1/2001  | SALES_REP  | 40000  | 0.3                  | 2         | 20           |
| 5          | ajay      | ghosala  | ghosala@ho<br>tmail.com | 9809888898  | 6/10/2002 | SALES_REP  | 30000  | 0.4                  | 2         | 20           |
| 6          | John      | Reddies  | John@gmail<br>.com      | 6878900989  | 6/10/2003 | M_per      | 50000  | NULL                 | NULL      | NULL         |

**Table Name : Department**

| DEPARTMENTID | DEPARTMENTNAME | MANAGERID | LOCATIONID |
|--------------|----------------|-----------|------------|
| 10           | Sales          | 1         | 1          |
| 20           | IT             | 2         | 2          |
| 30           | Marketing      | (null)    | 1          |

**Table Name : Location**

| LOCATIONID | CITY   |
|------------|--------|
| 1          | Pune   |
| 2          | Mumbai |

## Data manipulation language

- DML stands for Data Manipulation Language.
- DML statements are required when user wants to
  - Add new rows to the table
  - Update existing rows
  - Remove rows from the table
- A collection of DML statements that form a logical unit of work is called as Transaction.
- Below are the types of DML statements:
  - INSERT
  - DELETE
  - UPDATE

# INSERT Statement

- INSERT statement is used **to add new rows** in the table.

- Syntax:

INSERT into table [(column1 [,column2....])]

values (value, [value2...]);

- Above syntax can add only one row in the table at a time.
- If column list is omitted in the INSERT clause, then
  - insert value for each column in the corresponding table.
  - specify the values in the default order of the columns in the table.
  - Else, list the specific columns in the INSERT clause, in which data needs to be inserted
- Enclose the character and date values in single quotation marks.

## INSERT Statement

- Insert Data in all the columns of the table

Insert into Department

Values(40, 'HR', 4, 5);

- Insert data in specific columns

Insert into Department(department\_id, department\_name)

values(50,'ITIG');

-- In this case NULL values will be inserted in Manager\_id and Location\_id columns

- Insert data using special functions

insert into Employee (Employeeid, firstname, lastname,email, phonenumber, hiredate, jobid)

values(6,'Jogin','tale','jt@gmail.com',4343535353,SYSDATE,'Sales\_Rep');

# INSERT Statement

- Use of Substitution variables:
  - User can prefix '&' to a variable to prompt user for values at the run time.
  - User can save commands using the substitution variable in a file and then can execute the script file. Then user will be prompted to input a value for & substitution variable. And then the inputted values will be inserted in the table.
  - Substitution variables can be used along with SELECT, INSERT, UPDATE, DELETE statements.

- Insert data using Substitution variables

Insert into Department

values(&department\_id,'&department\_name',&manager\_id,&location\_id);

## INSERT Statement

- Common errors that can occur during execution of INSERT statement
  - Mandatory value missing for a not null column.
  - Duplicate value violates unique constraint
  - Foreign key constraint violated
  - Check constraint violated
  - Data type mismatch
  - Value too wide to fit in column



## UPDATE Statement

- UPDATE statement is used to modify existing rows in the table.
- User can update more than one row as well as more than one column at a time.

- Syntax:

Update Table

Set Column = value [, Column = value ...]

[ Where Condition ];

Where Condition' identifies rows to be updated. If omitted, all the rows in the table will be affected.

## UPDATE Statement

- Simple Update Query

```
Update employee  
set departmentid=40  
where employeeid=5;
```

- Update statement without Where clause

```
Update employee  
set departmentid=40;
```

-- If where condition is omitted, all the rows will be updated,

- Updating multiple columns

```
Update employee  
set departmentid= 40, managerid= 1  
where employeeid= 5;
```

## UPDATE Statement

- Explicit DEFAULT

The DEFAULT keyword can be used with UPDATE and INSERT statements to use the DEFAULT value specified for that column while table creation. If no DEFAULT value is specified then, NULL value is used.

Explicit DEFAULT in Update statement :

Update department

Set locationid = DEFAULT

Where departmentid=10;

## DELETE Statement

- DELETE statement is used to remove rows from the table.
- If user removes all rows from the table, only the data structure of the table remains.
- Syntax: DELETE [ from ] Table  
[ Where Condition ]

Where Condition' identifies rows to be deleted. If omitted, all the rows in the table will be affected.

Note: User can not delete a row that contains a primary key which is used as foreign key in another table. It will result in 'Integrity constraint' error

- Delete statement with Where clause  
Delete from employee  
where departmentid=10;
- Remove all the rows from the table  
Delete from employee;

## Integrity constraint error

- If user attempts to delete a record with a value that is tied to an integrity constraint, an error is returned.
- Example:

Delete from Department

Where departmentid=20;

- Above query will result in below error, as Employee table contains records with department id as 20.

ORA – 02292: integrity constraint violated(PK) – child record found

- User can not delete a row that contains a primary key that is used as a foreign key in another table. However, if corresponding referential integrity constraint contains ON DELETE CASCADE option, then the selected rows and its children are deleted from their respective tables.

## Session 3: Summary

With this we have come to an end of our third session where we discussed about meaning and Use of DML statements.

- At the end of Nugget 3, we see that you are now able to answer following questions:
  - Meaning and types of DML statements along with examples.
  - Explain: Use of Substitution variable with DML statements.



**Persistent**

**Thank you!**

Persistent University

