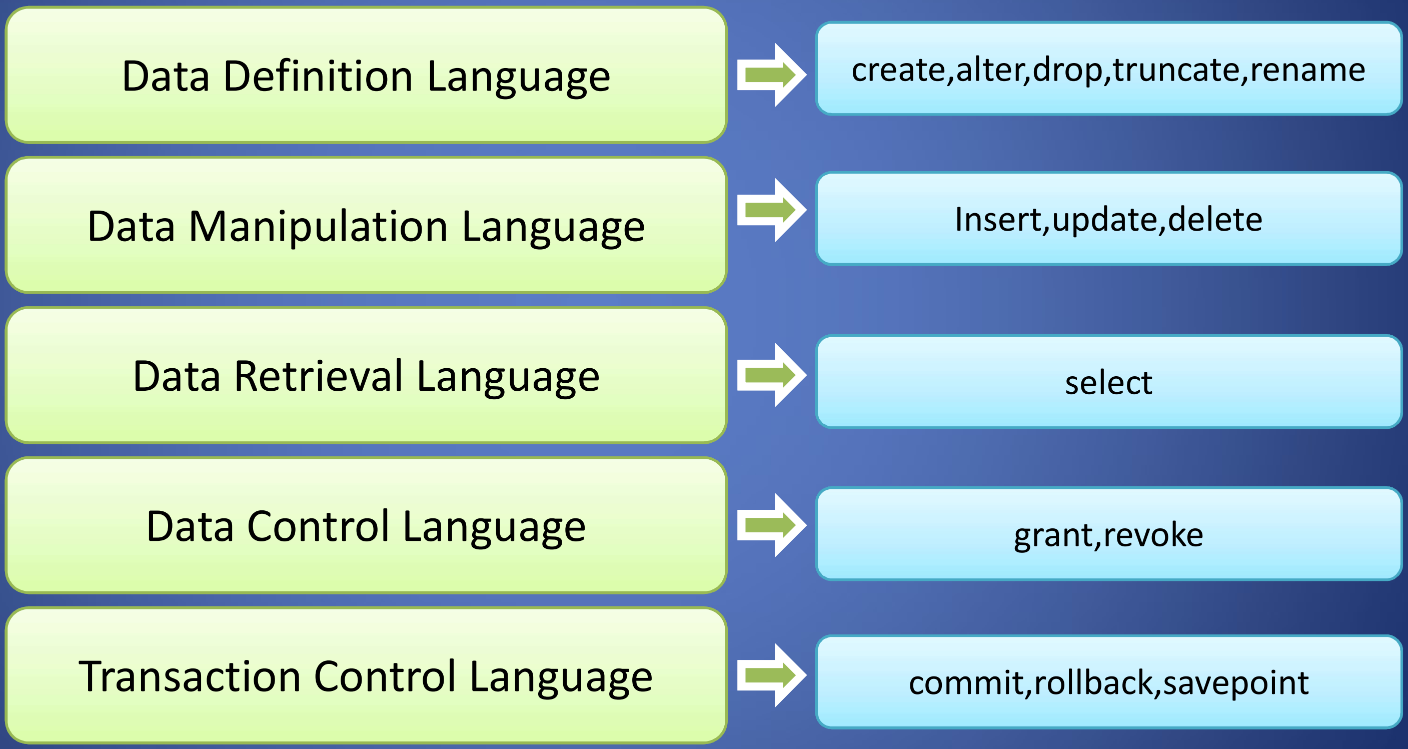
Get Started

❖ SQL DEVELOPER

Classification of SQL

- Data Definition Language
- Data Manipulation Language
- Data Retrieval Language
- Data Control Language
- Transaction Control Language

Classification of SQL



Data Definition Language → create,alter,drop,truncate,rename

Data Manipulation Language → Insert,update,delete

Data Retrieval Language → select

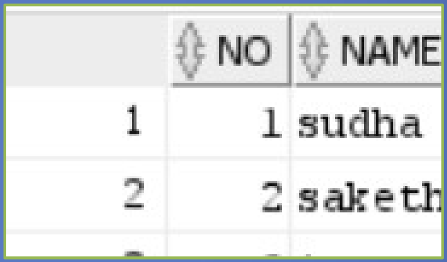
Data Control Language → grant,revoke

Transaction Control Language → commit,rollback,savepoint

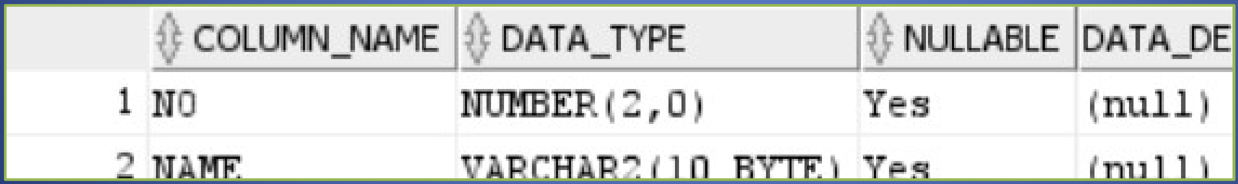
Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. “Students" ").

Tables contain records (rows) with data.



	NO	NAME
1	1	sudha
2	2	saketh



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DE
1	NO	NUMBER(2,0)	Yes	(null)
2	NAME	VARCHAR2(10 BYTE)	Yes	(null)

SQL Data Types

Each column in a database table is required to have a name and a data type.

An SQL developer must decide what type of data that will be stored inside each column when creating a table.

The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.

Oracle Datatypes

https://docs.oracle.com/cd/A58617_01/server.804/a58241/ch5.htm

Character data types

- ○ CHAR
- ○ NCHAR
- ○ VARCHAR2 and VARCHAR
- ○ NVARCHAR2
- ○ CLOB
- ○ NCLOB
- ○ LONG

NUMBER datatype

DATE datatype

Binary datatypes

- BLOB
- BFILE
- RAW
- LONG RAW

Creating Table

CREATE TABLE SYNTAX

Create table (col1 datatype1, col2 datatype2 …coln datatypen); Ex: SQL> create table student (no number (2), name varchar (10), marks number (3));

INSERT

INSERT This will be used to insert the records into table. We have two methods to insert.

- By value method
- By address method

INSERT

a) USING VALUE METHOD

Syntax: insert into <table_name) values (value1, value2, value3 …. Valuen);

Ex:

```
    SQL> insert into student values (1, ’sudha’, 100);
    SQL> insert into student values (2, ’saketh’, 200);
```

To insert a new record again you have to type entire insert command, if there are lot of records this will be difficult.

This will be avoided by using address method

INSERT

b) USING ADDRESS METHOD

Syntax:

```
insert into <table_name) values (&col1, &col2, &col3 …. &coln);
```

This will prompt you for the values but for every insert you have to use forward slash.

```
SQL> insert into insert into student values (&no, '&name', &marks);

Enter value for no: 1
Enter value for name: Jagan
Enter value for marks: 300

old 1: insert into student values(&no, '&name', &marks)
new 1: insert into student values(1, 'Jagan', 300)
```

INSERT

c) <u>INSERTING DATA INTO SPECIFIED COLUMNS USING VALUE METHOD</u>

Syntax:

```
        insert into <table_name)(col1, col2, … Coln) values (value1, value2, …. Valuen);
```

Ex:

```
SQL> insert into student (no, name) values (3, ’Ramesh’);
SQL> insert into student (no, name) values (4, ’Madhu’);
```