

# SWE1004: Database Management System

## DDL and DML Commands

# Syllabus

1.	SQL -Creating tables
2.	SQL- Inserting, deleting, updating tables, Alter table
3.	SQL -Querying table-simple queries
4.	SQL- Creating constraints
5.	SQL- Altering constraints
6	SQL- In built functions
7	SQL – Select statements(with different clauses)
8	SQL- Querying table-complex(nested, correlated)
9	SQL – Top N Queries , <u>catalog</u> Queries, views
10	PLSQL- block, cursor
11	PLSSQL- trigger
12	PLSQL- Function, Procedure
13	SQL-Creating and Querying-type, <u>varray</u> , nested table
14	API- Creating API for retrieving data from database
15	API- Creating API for executing procedure/function

# Topics to be Covered



- Introduction (Data, Database, DBMS)
- SQL
- DDL Commands
  - Create table
  - Alter table
  - Truncate table
  - Drop table
- DML Commands
  - Insert
  - Select
  - Update
  - Delete

# Introduction

- Database
  - Collection of Interrelated Data
- Database Management Systems
  - Set of programs used to manipulate database
- Relational Database
  - A database structured to recognize relations between stored items of information.
- Table – Real World Object
- Attribute / Field – Properties of a Table
- Tuple / Record – an ordered set of data constituting a record
- Domain / Column – Set of all values of an attribute

# Introduction

Employee Table

EMPNAME	EMPCITY	EMPSAL	DESIG	Tuple/Record
ram	chennai	9000	SL	
seetha	trichy	8000	SL	
raj	cbe	6000	Lect	
ganga	salem	8500	Lect	
ramu	tnvl	30000	AP	

Domain / Column

# Introduction



- SQL stands for Structured Query Language.
- Initially it was called as SEQUEL (Structured Query English Language)
- SQL lets you access and manipulate databases.
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.
- Although SQL is an ANSI/ISO standard, there are different versions of the SQL language.

# Structured Query Language (SQL)

- The ANSI standard language for the definition and manipulation of relational database.
- Includes data definition language (DDL), statements that specify and modify database schemas.
- Includes a data manipulation language (DML), statements that manipulate database content.

# Data Definition

- Metadata describes data about data.
- Metadata clearly explains the low level structure of data.
- Metadata highly dominates while the creation of database.
- CREATE statement is the universal command to construct a database.
- Data Definition is the pillar of database architecture



# Data Types

- Data types defines the type of value to be stored in the table.
- SQL data types can be broadly divided into following categories.
  - Numeric data types
  - Date and Time data types
  - Character and String data types
  - Unicode character string data types
  - Binary data types
  - Miscellaneous data types – clob(character large Object-4GB), blob (Binary large Object), xml, cursor, table etc.

# SQL Data Types

- String types
  - CHAR(n) – fixed-length character data, n characters long Maximum length = 2000 bytes
  - VARCHAR2(n) – variable length character data, maximum 4000 bytes
  - LONG – variable-length character data, up to 4GB. Maximum 1 per table
- Numeric types
  - NUMBER(p,q) – general purpose numeric data type
  - INTEGER(p) – signed integer, p digits wide
  - FLOAT(p) – floating point in scientific notation with p binary digits precision
- Date/time type
  - DATE – fixed-length date/time in dd-mm-yy form

22 Jan 2021

# SQL Environment

- Catalog
  - A set of schemas that constitute the description of a database
- Schema
  - The structure that contains descriptions of objects created by a user (base tables, views, constraints)
- Data Definition Language (DDL)
  - Commands that define a database, including creating, altering, and dropping tables and establishing constraints
- Data Manipulation Language (DML)
  - Commands that maintain and query a database
- Data Control Language (DCL)
  - Commands that control a database, including administering privileges and committing data

# DDL, DQL, DML & DCL

- Data Definition Language
- Data Query Language
- Data Manipulation Language
- Data Control Language

# DDL, DQL, DML & DCL

## SQL Commands

### DDL

CREATE

ALTER

DROP

RENAME

TRUNCATE

### DQL

SELECT

### DML

INSERT

UPDATE

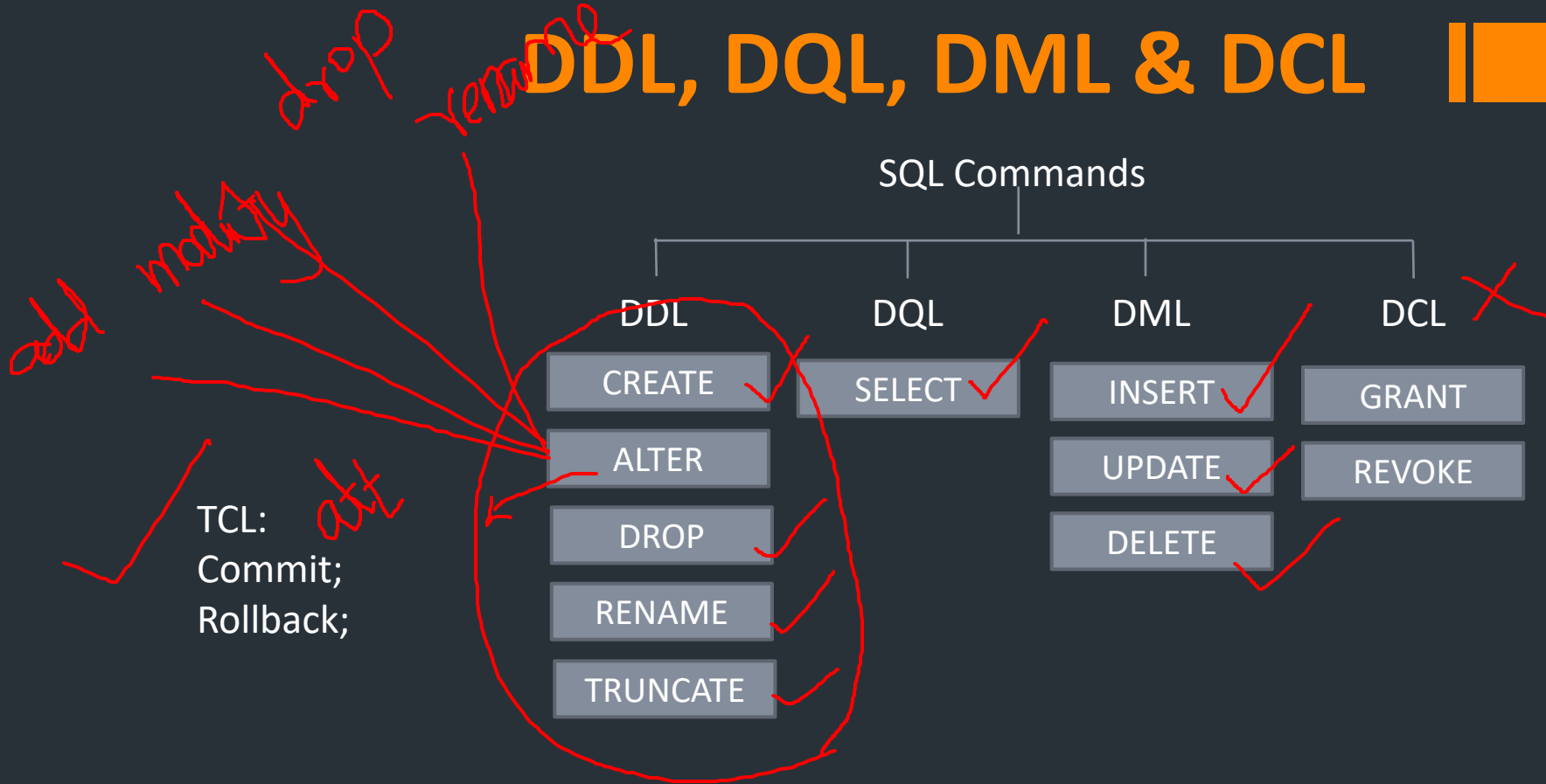
DELETE

### DCL

GRANT

REVOKE

TCL:  
Commit;  
Rollback;



# DDL, DQL, DML & DCL

## ■ DDL Syntax :

\*desc table\_name;

CREATE

```
CREATE TABLE table_name (  
  column1 datatype,  
  column2 datatype,  
  column3 datatype,  
  .... );
```

ALTER

```
ALTER TABLE table_name  
ADD column_name datatype;
```

DROP

```
DROP TABLE table_name;
```

# DDL, DQL, DML & DCL

- DDL Syntax :

RENAME

```
ALTER TABLE table_name  
RENAME TO new_table_name;
```

TRUNCATE

```
TRUNCATE TABLE table_name;
```

# DDL, DQL, DML & DCL



- **DQL Syntax :**

SELECT

SELECT column1, column2, ...  
FROM table\_name;

- **DML Syntax :**

INSERT

INSERT INTO TABLE\_NAME (column1,  
column2, column3,...columnN)]  
VALUES (value1, value2,  
value3,...valueN);



# DDL, DQL, DML & DCL



- DML Syntax (Cont.) :

UPDATE

```
UPDATE table_name  
SET column1 = value1,  
    column2 = value2, ...  
WHERE condition;
```

DELETE

```
DELETE FROM table_name  
WHERE condition;
```

# DDL, DQL, DML & DCL

- DCL Syntax :

GRANT

GRANT privilege\_name  
ON object\_name  
TO {user\_name | PUBLIC | role\_name}  
[WITH GRANT OPTION];

REVOKE

REVOKE privilege\_name  
ON object\_name  
FROM {user\_name | PUBLIC | role\_name}

# Basic Retrieval Queries in SQL

- SQL has one basic statement for retrieving information from a database: the SELECT statement
- SELECT statement is not the same as the SELECT operation of relational algebra
- The basic form of the SELECT statement is formed by three clauses SELECT, FROM, and WHERE
  - SELECT <attribute list>
  - FROM <table list>
  - WHERE <condition>

# DDL Commands



- Create table
- Describe
- Alter table
  - Add
    - Attribute
    - Constraints
  - Rename
    - Table
    - Attribute name
- Modify
  - Data type
  - Size of attribute
  - Constraints
- Drop
  - Attribute
  - Constraints
- Truncate table
- Drop table

# DDL Commands

- Create Table

```
create table student(name varchar2(15), rollno number(8), dept varchar2(5),  
doj date);
```

- Describe Table

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
NAME		VARCHAR2 (15)
ROLLNO		NUMBER (8)
DEPT		VARCHAR2 (5)
DOJ		DATE

# DDL Commands

- Alter Table

- Add

```
SQL> alter table student add regno number(15);
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
NAME		VARCHAR2 (15)
ROLLNO		NUMBER (8)
DEPT		VARCHAR2 (5)
DOJ		DATE
REGNO		NUMBER (15)

# DDL Commands

- Alter Table
- Modify - Data Type and Attribute Size
- Modify - Data Type

```
SQL> alter table student modify rollno varchar2(7);
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
NAME		VARCHAR2 (15)
ROLLNO		VARCHAR2 (7)
DEPT		VARCHAR2 (5)
DOJ		DATE
REGNO		NUMBER (15)

# DDL Commands

- Alter Table
- Modify - Attribute Size

```
SQL> alter table student modify rollno varchar2(8);
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
NAME		VARCHAR2 (15)
ROLLNO		VARCHAR2 (8)
DEPT		VARCHAR2 (5)
DOJ		DATE
REGNO		NUMBER (15)



# Insert Command

```
SQL> insert into student values('kalai','vit1001','cse','01-aug-2021',4001);
```

```
1 row created.
```

```
SQL> insert into student values('Pon','vit1002','cse','01-jul-2021',4002);
```

```
1 row created.
```

```
SQL> select * from student;
```

NAME	ROLLNO	DEPT	DOJ	REGNO
-----	-----	-----	-----	-----
kalai	vit1001	cse	01-AUG-21	4001
Pon	vit1002	cse	01-JUL-21	4002

# DDL Commands

- Alter Table
- Modify - Drop

```
SQL> alter table student drop column regno number(15);
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
NAME		VARCHAR2 (15)
ROLLNO		VARCHAR2 (8)
DEPT		VARCHAR2 (5)
DOJ		DATE

# DDL Commands

- Alter Table
- Modify - Drop

```
SQL> alter table student drop column regno number(15);
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
NAME		VARCHAR2 (15)
ROLLNO		VARCHAR2 (8)
DEPT		VARCHAR2 (5)
DOJ		DATE

# DDL Commands

- Truncate Table

```
SQL> truncate table student;
```

```
Table truncated.
```

```
SQL> select * from student;
```

```
no rows selected
```

```
SQL> desc student
```

Name	Null?	Type
NAME		VARCHAR2 (15)
ROLLNO		VARCHAR2 (8)
DEPT		VARCHAR2 (5)
DOJ		DATE
REGNO		NUMBER (15)

# DDL Commands

- Drop Table

```
SQL> drop table student;
```

```
Table dropped.
```

```
SQL> select * from student;  
select * from student  
          *
```

```
ERROR at line 1:
```

```
ORA-00942: table or view does not exist
```

```
SQL> desc student
```

```
ERROR:
```

```
ORA-04043: object student does not exist
```

# DDL Commands

- Drop Table

```
SQL> drop table student;
```

```
Table dropped.
```

```
SQL> select * from student;  
select * from student  
          *
```

```
ERROR at line 1:
```

```
ORA-00942: table or view does not exist
```

```
SQL> desc student
```

```
ERROR:
```

```
ORA-04043: object student does not exist
```

# DDL Commands

- Rename Table / Rename Attribute / Copy a Table

- copy a table

```
create table table_name as select * from table_name;
```

- Rename a column name

```
alter table table_name rename column old_name to new_name;
```

- Rename a table

```
alter table table_name rename to new_name;
```

- Examples:

```
alter table student rename column branch to course;
```

```
alter table student rename to students;
```

# DML Commands

```
SQL> create table player(name varchar2(20), country varchar2(15),  
    matches number(3), runs number(4), jersy number(2));
```

Table created.

## Insert Command

Method 1 - Inserting Single records

```
SQL> insert into player values('sachin','india',98,999,9);
```

1 row created.



# DML Commands

## Method 2 - Inserting Multiple records

```
SQL> insert into player  
      values ('&name', '&country', &matches, &runs, &jersy) ;
```

```
Enter value for name: shewag
```

```
Enter value for country: india
```

```
Enter value for matches: 15
```

```
Enter value for runs: 750
```

```
Enter value for jersy: 5
```

```
old   1: insert into player  
      values ('&name', '&country', &matches, &runs, &jersy)
```

```
new   1: insert into player values ('shewag', 'india', 15, 750, 5)
```

```
1 row created.
```

# DML Commands

```
SQL> /
Enter value for name: gilchrist
Enter value for country: australia
Enter value for matches: 50
Enter value for runs: 5000
Enter value for jersy: 3
old 1: insert into player
      values('&name','&country',&matches,&runs,&jersy)
new 1: insert into player
      values('gilchrist','australia',50,5000,3)

1 row created.
```

# DML Commands

Method 3 - Inserting Values for selected attributes

```
SQL> insert into player(name,country,matches)
      values('pathani','india',02);
```

1 row created.

# DML Commands

## Select Command

Selecting the overall contents (tuples) of the relation

```
SQL> select * from player;
```

NAME	COUNTRY	MATCHES	RUNS	JERSY
sachin	india	98	999	9
shewag	india	15	750	5
gilchrist	australia	50	5000	3
pathani	india	2		

4 rows selected.

# DML Commands

Selecting domain values

```
SQL> select name,country from player;
```

NAME	COUNTRY
sachin	india
shewag	india
gilchrist	australia
pathani	india

4 rows selected.

# DML Commands

Selecting values by avoiding repetitions and duplications

```
SQL> select distinct country from player;
```

```
COUNTRY
```

```
-----
```

```
australia
```

```
india
```

# DML Commands

Selecting tuple values based on conditions

```
SQL> select * from player where matches>=40;
```

NAME	COUNTRY	MATCHES	RUNS	JERSY
sachin	india	98	999	9
gilchrist	australia	50	5000	3

# DML Commands

Selecting domain values based on conditions

```
SQL> select name,runs from player where runs>=1000;
```

NAME	RUNS
gilchrist	5000



# DML Commands

- Selecting tuple values based on sorting in ascending order

```
SQL> select * from player order by matches;
```

- Selecting tuple values based on sorting in ascending order by specifying asc

```
SQL> select * from player order by matches asc;
```

- Selecting tuple values based on sorting in descending order by specifying desc

```
SQL> select * from player order by matches desc;
```

- Selecting tuple values based on sorting with both asc and desc

```
SQL> Select * from player order by matches asc, country desc;
```

# DML Commands

- Selecting tuple values by using string constants

- Selecting tuple values whose name starts with 's'

```
SQL> select * from player where name like 'sh%'; (starts with s
```

- Selecting tuple values whose 5 character name starts with 'd' and ends with 'i'

```
SQL> select * from player where name like 'd__i';
```

- Selecting tuple values whose name has character 'c'

```
SQL> select * from player where name like '%c%'; (any where)
```

- Renaming domain name

```
SQL> select name player, country nation from player;
```

# DML Commands

- Update Command

- Change the runs as 10 whose jersey number is 10

```
SQL> update player set runs=10 where jersey=10;
```

- Change the runs as 10 whose jersey number is null

```
SQL> update player set runs=10 where jersey is null;
```

```
3 rows updated.
```

# DML Commands

- Delete Command

- Deleting based on conditions

```
SQL> delete from player where jersy is null;
```

- Deleting overall contents from the relation

```
SQL> delete from player;
```

```
SQL> select * from player;
```

```
no rows selected
```



**Thank You...**