SQL

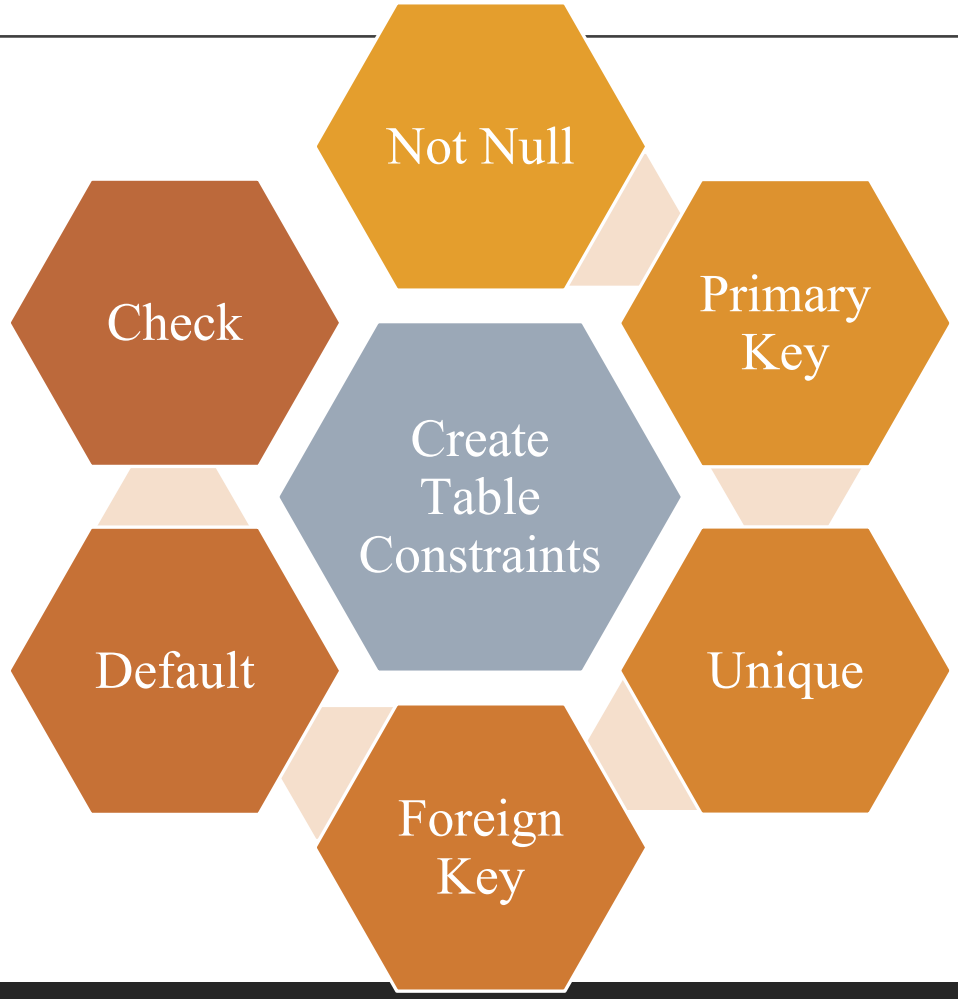
INTRODUCTION

LECTURE 10

Contents to be covered:

- Create Table Contraints
  - Not Null
  - Check
  - Default Key
- Data Manipulation Language (DML)
- DML commands
  - Insert into
  - Update
  - Delete
  - Select

Create Table Constraints



Note : Adding constraints to a table enables the database system to enforce data integrity.

Not Null Constraint

```
CREATE TABLE Employee (
    Employee_Id    Integer,
    Employee_Name  VARCHAR2(20) NOT NULL,
    Employee_Address  VARCHAR2(20) NOT NULL,
);
```

Check Constraint

- We can define a CHECK constraint in a single column as well as in a table.
- If we define a CHECK constraint on a single column it allows only certain range of values for this column.
- If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

Check Constraint

```
CREATE TABLE Employee (
    Employee_Id    Integer NOT NULL CHECK(Employee_Id >0),
    Employee_Name  VARCHAR(20),
    Employee_City  VARCHAR(20),
);
```

Check Constraint

```
CREATE TABLE Employee (
    Employee_Id    Integer NOT NULL,
    Employee_Name  VARCHAR(20),
     Employee_City VARCHAR(20),
    Constraints emp_check CHECK(Employee_Id >0)
);
```

Check Constraint(Multiple columns)

```
CREATE TABLE Employee (
    Employee_Id    Integer NOT NULL,
    Employee_Name  VARCHAR(20) NOT NULL,
    Employee_City VARCHAR(20),
        Employee_Age Integer,
    Constraints emp_check CHECK(Employee_Age >20 AND
Employee_City='Mumbai')
);
```

Check on Alter and Drop Command

- **ALTER TABLE Employee**
  ADD CHECK (Emp_Age>=20);

- **ALTER TABLE Employee**
  ADD CONSTRAINT CHK_EmpAge CHECK (Emp_Age>=20 A
  NDCity='Mumbai');

- **ALTER TABLE Employee DROP Check CHK_EmpAge;**

Default Constraints

- This constraint is used to insert a default value into a column.
- The default value will be added to all new records, if no other value is specified.

```
CREATE TABLE Employee (
    Employee_Id   Integer,
    Employee_Name  VARCHAR(20) NOT NULL,
    Employee_ City VARCHAR(20) DEFAULT 'Mumbai');
```

Default with Alter or Drop Command

ALTER TABLE Employee ALTER City SET DEFAULT 'Mumbai';

ALTER TABLE Employee ALTER City DROP DEFAULT;

Data Manipulation Language

DML Command: Insert into

- This command is used to insert the records into a table.
- The attribute value should be inserted in an order.
- String data must be enclosed in single quotes.
- Numbers are not quoted

DML Command: Insert into

Syntax:

```
INSERT INTO <Tablename> (column, …, column)
VALUES (value, …, value);
```

or

```
INSERT INTO <Tablename> VALUES (value, …, value);
```

Example: Insert into

```
INSERT INTO Employee
VALUES (333,'Jane','Mumbai');
```

Relation: Employee

Employee_id	Employee_Name	Employee_city
111	David	Delhi
222	Peter	Delhi

Relation: Employee (After Insertion)

Employee_id	Employee_Name	Employee_city
111	David	Delhi
222	Peter	Delhi
333	Jane	Mumbai

Example: Insert into

Records can be fetched from another table for insertions.

For example: If we want to insert the records present in Table2 into Table1. command will be given as follows:

```
INSERT INTO Table1
        Select * from Table2
```

DML Command: Update

DML Command: Update

This command is used to modify the data of the table.

Syntax:

```
UPDATE <table name>
SET <column> = <value>
WHERE <selection condition>;
```

Example: Update

```
UPDATE Employee
SET  Employee_city='Kolkata'
WHERE Employee_id=111 and Employee_Name='David';
```

Relation: Employee

Employee_id	Employee_Name	Employee_city
111	David	Delhi
222	Peter	Delhi
333	Jane	Mumbai

Relation: Employee (After Insertion)

Employee_id	Employee_Name	Employee_city
111	David	Kolkata
222	Peter	Delhi
333	Jane	Mumbai

DML Command: Delete

DML Command:Delete

The Delete command is used to delete the records in a table.

Syntax:

```
DELETE FROM <table_name>
WHERE <condition>
```

Example: Delete

For example: We want to delete all employees from city Delhi

```
DELETE FROM Employee
WHERE Employee_city='Delhi'
```

Relation: Employee

Employee_id	Employee_Name	Employee_city
111	David	Delhi
222	Peter	Delhi
333	Jane	Mumbai

Relation: Employee (After Deletion)

Employee_id	Employee_Name	Employee_city
333	Jane	Mumbai

Example: Delete

For example: We want to delete all employees

```
DELETE FROM Employee;
```

Relation: Employee

Employee_id	Employee_Name	Employee_city
111	David	Delhi
222	Peter	Delhi
333	Jane	Mumbai

Relation: Employee (After Deletion)

Employee_id	Employee_Name	Employee_city

DML Command: Select

DML Command: Select

The Select command is used to extract the few records from the table based on application requirement.

Syntax:

```
SELECT <column name>
FROM <Table name>
WHERE <condition>;
```

Example: Select

Example Query 1 : Display all the employee records

Note: An asterisk in the select clause denotes “all attributes”

```
SELECT * FROM Employee;
```

Example Query 2: Display the employees details belonging to Delhi

```
SELECT * FROM Employee WHERE Employee_city=‘Delhi’;
```

Example: Select

- To display few attributes corresponding to query

Example Query 3 : Display city of all the employees

```
SELECT Employee_city FROM Employee;
```

Example Query 4: Display city of the employees whose id is 222.

```
SELECT Employee_city FROM Employee WHERE
Employee_id=222;
```

Example: Select

Query3 will display the city of all the employees in which duplicate data might appear as employee relation can have more than one employee from same city.

To eliminate the duplicacy, distinct keyword is used

Example Query 5 : Display the city of all the employee with no duplicates.

```
SELECT distinct Employee_city * FROM Employee;
```

Example: Select having arithmetic expressions

```
SELECT Emp_name, Emp_Sal, Emp_sal+500 Employee;
```

Result is:

Emp_Name	Emp_Sal	Emp_Sal+500
David	10000	10500
Peter	15000	15500
Jane	20000	20500

```
SELECT Emp_name, Emp_Sal, 10*Emp_sal+500
Employee where Emp_Sal<12000;
```

Result is:

Emp_Name	Emp_Sal	Emp_Sal+500
David	10000	100500

ALIAS

SQL aliases are used to temporarily rename a table or a column heading.

Column Name Alias

The syntax is:

SELECT column AS column_alias FROM table

Table Name Alias

The syntax is:

SELECT column FROM table AS table_alias

Note: Mostly used with join queries.

Example: Column Alias

Employee_id	Employee_Name	Employee_city
111	David	Delhi
222	Peter	Delhi
333	Jane	Mumbai

```
SELECT Emp_name as name, Emp_city as city FROM Employee;
```

Result is:

Name	City
David	Delhi
Peter	Delhi
Jane	Mumbai

Practice SQL Queries:

Consider the following Schema :
Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)
Works(*eid:* integer, *did:* integer, *pcttime:* integer)
Dept(*did:* integer, *dname:* string, *budget:* real, *managerid:* integer)
Where *Emp* describes employees, *Dept* describes departments and *Works* describes the percentage of time an employee works with a given department.

Give an answer to the following statements:
1. Identify key attributes of all the relations.
2. Identify the attributes that will follow referential integrity constraints.
3. How to add a constraint that the salary of all employees should be greater than 10000
4. How to add a constraint to ensure that employee name should not be left blank.
5. How to add a constraints to ensure the employee age lies between 30 to 50.
6. Write an SQL statement to give every employee a 10 percent raise.
7. Write an SQL statement to delete the Toy department. Given the referential integrity constraints you choose for this schema, what happens when this statement is executed?

MySQL

Introduction to MySQL

- MySQL is an open source RDBMS

- Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

- MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability.

Reading Suggestion : MySQL, Wikipedia.

MySQL

- A MySQL database server contains many databases.
- Each database consists of one or more tables. A table is made up of columns (or fields) and rows (records).
- The SQL keywords and commands are NOT case-sensitive.
- The *names* or *identifiers* (database names, table names, column names, etc.) are case-sensitive in some systems, but not in other systems. Hence, it is best to treat *identifiers* as case-sensitive.

My SQL

SHOW DATABASES: This is command to list all existing databases in the server

The databases "mysql", "information_schema" and "performance_schema" are system databases used internally by MySQL.

MySql

Database changed

MySQL

For creating the database in the MySQL

mysql> CREATE DATABASE dbs_name;
Query OK, 1 row affected

Ones database is created , the tables can be created within that database. To create the tables within a database we need to execute the USE command

mysql> USE dbs_name;
Database changed

MySQL

To see the tables in the current database - dbs_name

```
Mysql>SHOW TABLES;
Empty set (0.00 sec)
```

Create a table, student having column student id, student name, student course

```
Mysql> CREATE TABLE Student {
stuid integer primary key,
stu_name varchar(20),
stu_course varchar(10)};
Query OK, 0 rows affected (0.08 sec)
```

MySQL

Mysql>SHOW TABLES;

- **To view all the attributes of a table**

Mysql>DESC STUDENT;

MySQL

For deleting the database from the MySQL

```
mysql> DROP DATABASE dbs_name;
Query OK, 0 rows affected
```