SQL
DATA MANIPULATION

Prepared By: Dr. Vipul Vekariya
SQL
DATA MANIPULATION

- SQL DATA TYPES
- CREATE CLAUSE
- SELECT CLAUSE
- ORDERED BY CLAUSE
- AS CLAUSE
Basic Data Types of SQL

- CHAR
- NUMERIC
- VARCHAR/VARCHAR 2
- DATE
CHAR(size)

- Used to store a character string value
- Fixed-length character string stored in cell
- Size defines the number of characters stored
- Maximum 255 characters stored

**Example**

Name CHAR(60);
**VARCHAR(size)/VARCHAR2(size)**

- Used to store Variable Length string
- Alphanumeric String Stored
- Maximum size is up to 4000-Character

**EXAMPLE:**
- Name VARCHAR2(20);
DATE

• Used to represent Date and Time
• Standard format is dd-mon-yy (21-JUL-08)
• To Enter other format special function is used.
• Default Date is 1st Day of Current Month.

• Time is Stored in 24-Hour format.
• Default Time is 12:00 AM
NUMBER(p,s)

- Used to store Fixed or Floating Point Values.
- P → Precision (Indicate Digits)
- S → Scale (Indicate Digits After Floating Point)
- Maximum 38-Digits are Stored.

**Example:**
- Rupee NUMBER(7,2).
Rules of SQL

- SQL Statements are start with Verb.( example SELECT)
- Each verb followed by number of clause. Example(FROM, WHERE).
- A space separates clause like DROPTABLE EMP;
- SQL Statements are end with Semicolon( ;).
- SQL Parameters are separated by Comma( ,).
- Statement may be split across lines but keyword may not.
- Reserved verb can not used as Identifier.
- Identifier must start with Alphabet & should not more than 30 characters.
- Comments may be enclosed between /* */ symbol.
Basic Structure of SQL

- Consists of three clauses:
  (i) Select
    - Used to list the attributes desired in the result of a query.
  (ii) From
    - Lists the relations to be scanned in the evaluation of the expression.
  (iii) Where
    - Consists of a predicate involving attributes of the relations that appear in the from clause.
CREATE Clause

• Used to Create Table in Database.

• Syntax:
  • CREATE TABLE <table_name>(<Colname1> <DataType> (<Size>),
    (<Colname1> <DataType> (<Size>));

• Example:
  • CREATE TABLE Student_Master
    (Roll_No NUMBER(10),
     Name VARCHAR2(50),
     marks number(5,2));
Rules For Creating Table

- Name of Data Table begin with Alphabet
- A-Z, a-z, 0-9 characters are used
- Must be less than 30-character.
- Special Symbols _,#,$ are used.
- Reserved words like CREATE, SELECT, WHERE can not be used.
- Each columns of Table must contain DataType.
INSERT INTO Clause

• Used to Insert Data into Table.

• Syntax:
  • `INSERT INTO <tablename> (colname1,colname2) VALUES (<expression1>,<expression2>);`

• Example:
  • `INSERT INTO Student_Master (Rollno,Name) VALUES (123,‘XYZ’);`
SELECT Clause

• Used to View or Retrieve Data of Table.

• Syntax:
  - SELECT <col1>,<col2>,…<coln> FROM <Tablename>;

• Example:
  - SELECT Rollno,Name FROM Student_Master;
Filtering of Data

- All rows and All Columns.
- Selected Columns and All Rows.
- Selected Rows and All Columns.
- Selected Rows and Selected Columns.
All Rows and All Columns

- Astric (*) sign is used With Select Clause.
- Syntax:
  - `SELECT * FROM <tablename>;`
- Example:
  - `SELECT * FROM Student_Master;`

```
Student_Master

<table>
<thead>
<tr>
<th>ROLLNO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>XYZ</td>
</tr>
<tr>
<td>02</td>
<td>PQR</td>
</tr>
<tr>
<td>03</td>
<td>TUV</td>
</tr>
</tbody>
</table>
```
Selected Column All Rows

- Retrieve only Specific column from Table.
- **Syntax:**
  - `SELECT <col1>,<col2> FROM <tablename>;`
- **Example:**
  - `SELECT Rollno From Student_Master;`

<table>
<thead>
<tr>
<th>Rollno</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
</tr>
<tr>
<td>02</td>
</tr>
<tr>
<td>03</td>
</tr>
</tbody>
</table>
Condition used with the SELECT Clause
For Condition Where Clause is used
Condition is a combination of Relational operator and conditional operator.
BETWEEN and NOT BETWEEN Clause are used with the Where Clause
WHERE Clause

**Syntax:**
- SELECT * FROM <tablename>
  WHERE <condition>

**Example:**
- SELECT * FROM Student_Master
  WHERE Rollno = 01;

- SELECT * FROM Student_Master
  WHERE Rollno between 10 and 20;
ORDER BY Clause

- Used to Sort data of Table
- Used with Select Clause
- Syntax:
  - SELECT * FROM <tablename>
    ORDER BY <colname> <[sort_order]>;

- Example:
  - SELECT * FROM Student_Master
    ORDER BY Rollno DESC;

<table>
<thead>
<tr>
<th>Rollno</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>STU</td>
</tr>
<tr>
<td>02</td>
<td>PQR</td>
</tr>
<tr>
<td>01</td>
<td>XYZ</td>
</tr>
</tbody>
</table>
DISTINCT Clause

- Used with Select clause to eliminate duplication of row.

**Syntax:**
- SELECT DISTINCT <colname> FROM <tablename>;

**Example:**
- SELECT DISTINCT Occupation FROM Customer_Master;
DELETE CLASUE

- Used to Remove Rows of Table.

- Works in Two method.
  - Remove All rows of Table
  - Remove Selected rows of Table.
CREATING TABLE FROM A TABLE

• Syntax:

CREATE TABLE <tablename>(<Colname1>,<colname2>)
AS SELECT <colname1>,<colname2> FROM <tablename>;

Example:

CREATE TABLE ACCT_DTLS(ACCT_NO,BRANCH_NO,BALANCE)
AS SELECT ACCT_NO,BRANCH_NO,CURBAL FROM ACCT_MSTR;
Rename Operation

- As Clause is used to rename name of column as well as name of Table temporary in output.

Syntax:
- Old_name  as  new_name;

Example:
- SELECT Rollno as Regno
  FROM Student_Master as Student_Detail

<table>
<thead>
<tr>
<th>Student_Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regno</td>
</tr>
<tr>
<td>01</td>
</tr>
<tr>
<td>02</td>
</tr>
<tr>
<td>03</td>
</tr>
</tbody>
</table>
Remove all Rows of Table

- **Syntax**
  
  ```sql
  DELETE FROM <Table_Name>;
  ```

- **Example**
  
  ```sql
  DELETE FROM STUDENT_MASTER;
  ```

<table>
<thead>
<tr>
<th>ROLL_NO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Z</td>
</tr>
<tr>
<td>4</td>
<td>P</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROLL_NO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Remove Specified Row of Table

- **Syntax**
  - `DELETE FROM <TABLE_NAME>
    WHERE <CONDITION>`

- **Example**
  - `DELETE FROM STUDENT_MASTER
    WHERE ROLL_NO = 3;`

<table>
<thead>
<tr>
<th>ROLL_NO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROLL_NO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
</tr>
</tbody>
</table>
UPDATE CLAUSE

- Used to change or modify Table Data
- Used to change contents of Data Table

- Two Methods:
  - All rows Update at a Time
  - Selected rows Update at a Time
Update of All Rows

- **Syntax:**
  - `UPDATE <TABLE_NAME>`
  - `SET <COL_NAME> = <EXPERSSION>;

- **Example:**
  - `UPDATE SALARY`  
    `SET AMOUNT = 25000;`

<table>
<thead>
<tr>
<th>NAME</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>5000</td>
</tr>
<tr>
<td>Y</td>
<td>6000</td>
</tr>
<tr>
<td>Z</td>
<td>7000</td>
</tr>
<tr>
<td>W</td>
<td>8000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>25000</td>
</tr>
<tr>
<td>Y</td>
<td>25000</td>
</tr>
<tr>
<td>Z</td>
<td>25000</td>
</tr>
<tr>
<td>W</td>
<td>25000</td>
</tr>
</tbody>
</table>
UPDATE OF SELECTED ROWS

- **SYNTAX:**
  - UPDATE <TABLE_NAME>
    SET <COL_NAME> = <EXPRESSION>
    WHERE <CONDITION>;

- **EXAMPLE:**
  - UPDATE SALARY SET AMOUNT = 5000
    WHERE NAME = ‘X’;

<table>
<thead>
<tr>
<th>NAME</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>2500</td>
</tr>
<tr>
<td>Y</td>
<td>2500</td>
</tr>
<tr>
<td>Z</td>
<td>2500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>2500</td>
</tr>
<tr>
<td>Y</td>
<td>2500</td>
</tr>
<tr>
<td>Z</td>
<td>2500</td>
</tr>
<tr>
<td>X</td>
<td>5000</td>
</tr>
<tr>
<td>Y</td>
<td>2500</td>
</tr>
<tr>
<td>Z</td>
<td>2500</td>
</tr>
</tbody>
</table>
ALTER TABLE CLAUSE

• Used to modify the structure of Table.

• Used to
  • Add Column in Existing Table.
  • Delete Column From Existing Table.
  • Modify Column of Existing Table.
ADD Column in Table

• Syntax:
  • ALTER TABLE <TABLE_NAME>
    ADD (<NEW_COLNAME> <DATATYPE> <(SIZE)>);

• Example:
  • ALTER TABLE STUDENT_MASTER
    ADD (CITY VARCHAR2(10), STATE VARCHAR2(10));

<table>
<thead>
<tr>
<th>ROLL_NO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROLL_NO</th>
<th>NAME</th>
<th>CITY</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DELETE COLUMN FROM TABLE

- **SYNTAX:**
  - ALTER TABLE `<TABLE_NAME>`
    DROP COLUMN `<COLUMN_NAME>`;

- **EXAMPLE:**
  - ALTER TABLE SALARY
    DROP COLUMN ADDRESS;

<table>
<thead>
<tr>
<th>NAME</th>
<th>AMOUNT</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>25000</td>
<td>STREET</td>
</tr>
<tr>
<td>Y</td>
<td>7000</td>
<td>L.A.</td>
</tr>
<tr>
<td>Z</td>
<td>90000</td>
<td>V.T.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>25000</td>
</tr>
<tr>
<td>Y</td>
<td>7000</td>
</tr>
<tr>
<td>Z</td>
<td>90000</td>
</tr>
</tbody>
</table>
MODIFY COLUMN OF TABLE

**SYNTAX:**

- `ALTER TABLE <TABLE_NAME>
  MODIFY (<COL_NAME> <NEW_DATATYPE <(SIZE)>);

**EXAMPLE:**

- `ALTER TABLE STAFF_MASTER
  MODIFY NAME VARCHAR2(100);`
RESTRICTIONS OF ALTER TABLE

- User can not change the name of table
- User can not change the name of the column of specified table
- User can not decrease the size of column if the data is exist in column
RENAME CLAUSE

- USED TO Change the Name of Existing Table

- SYNTAX:
  - RENAME <TABLE_NAME> TO <NEW_TABLE_NAME>;

- EXAMPLE:
  - RENAME STAFF_MASTER TO FACULTY_MASTER;
TRUNCATE TABLE CLAUSE

- Used to Empty Table completely.

SYNTAX:
- TRUNCATETABLE <TABLE_NAME>;

EXAMPLE:
- TRUNCATETABLE STUDENT_MASTER;
DROP TABLE CLAUSE

- Used to Destroy table from Database.

**SYNTAX:**
- DROP TABLE <Table_Name>;

**EXAMPLE:**
- DROP TABLE Loan_Transaction;
<table>
<thead>
<tr>
<th>TRUNCATE TABLE CLAUSE</th>
<th>DELETE CLAUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop the Table and then Recreate the Table</td>
<td>Drop Rows of Table one by one</td>
</tr>
<tr>
<td>Faster than Delete Clause</td>
<td>Slow compare to Truncate Table Clause</td>
</tr>
<tr>
<td>No of Deleted rows are not retrieve</td>
<td>No of Deleted rows are retrieved.</td>
</tr>
<tr>
<td>Not a Safe-Transaction</td>
<td>Safe-Transaction.</td>
</tr>
</tbody>
</table>
DESCRIBE CLAUSE

- Represent structure of Table in Database.
- Describe datatype of column and no of columns available in Table.

SYNTAX:
- DESCRIBE <Table_Name>;}
Search Conditions

- The Comparison Test
- The Range Test
- The Set Membership Test
- Compound Search Conditions
- The Pattern Matching Test
- The Null Value Test
Comparison Test

- Compare value of One Expression to the Value of another Expression.

- Six comparison Operators are used
  - $=$ (Equal)
  - $<>$(Not Equal)
  - $<$ (Less than)
  - $>$ (Greater Than)
  - $<=$ (Less than or Equal)
  - $>=$(Greater than or Equal)
Range Test

- It Checks whether a data value lies between Two specified values.

- BETWEEN operator is Used for Range Test

- Three Expression are there
  - 1st Expression defined the value to be Tested
  - 2nd & 3rd Exp. Defines the range to be checked

- Syntax:
  - SELECT EMPLOYEEIDNO
    FROM EMPLOYEESTATISTICSTABLE
    WHERE SALARY BETWEEN 30000 AND 50000;

  - SELECT EMPLOYEEIDNO
    FROM EMPLOYEESTATISTICSTABLE
    WHERE SALARY NOT BETWEEN 30000 AND 50000;
Set Membership Test

- Test Whether a data value matches one of a list of target values.
- IN Operator is used
- Syntax:
  - Test-Exp. IN (constant)
  - Test-Exp. NOT IN (constant)
- Example:
  - SELECT * FROM STUDENT_MASTER
    WHERE ROLL_NO IN (11,12,13);
  - SELECT EMPLOYEEIDNO
    FROM EMPLOYEEESTATISTICSTABLE
    WHERE POSITION NOT IN ('Manager', 'Staff');
Compound Condition

- **AND, OR and NOT Operators are used**
  - The *AND* operator joins two or more conditions, and displays a row only if that row's data satisfies **ALL** conditions listed (i.e. all conditions hold true). For example, to display all staff making over $40,000, use:

  ```sql
  SELECT EMPLOYEEIDNO
  FROM EMPLOYEESTATISTICSTABLE
  WHERE SALARY > 40000 AND POSITION = 'Staff';
  ```

- **Example:**

  ```sql
  SELECT EMPLOYEEIDNO
  FROM EMPLOYEESTATISTICSTABLE
  WHERE SALARY > 40000 AND POSITION = 'Staff';
  ```
OR Operator

- The OR operator joins two or more conditions, but returns a row if ANY of the conditions listed hold true. To see all those who make less than $40,000 or have less than $10,000 in benefits, listed together, use the following query:

Example:

```
SELECT EMPLOYEEIDNO
    FROM EMPLOYEEESTATISTICSTABLE
    WHERE SALARY < 40000 OR BENEFITS < 10000;
```

```
SELECT EMPLOYEEIDNO
    FROM EMPLOYEEESTATISTICSTABLE
    WHERE POSITION = 'Manager' AND SALARY > 60000 OR BENEFITS > 12000;
```
Null Value Test

- Used to check whether null values are present in the column or not.
- IS NULL operator is Used.

**Syntax:**
- WHERE Column-name IS NULL
- WHERE Column-name IS NOT NULL

**Example:**
- SELECT NAME FROM SALES
  WHERE SALE IS NOT NULL
The Pattern Matching Test

- Used to compare one string value to another string value.

- LIKE operator is used for pattern matching

- Two Wildcard character are used
  - % (Percentage Sign) - match any substring
  - _ (Underscore Sign) - match any character
Like with %

- Allows to match any string of any length including zero length string

**Syntax:**
- Column-name LIKE 'patternstring %';

**Example:**

SELECT EMPLOYEEIDNO
FROM EMPLOYEEADDRESSTABLE
WHERE LASTNAME LIKE 'S%';
LIKE with _

- Used to match a Single Character.

- Syntax:
  - Column-name LIKE 'character _';

- Example:
  - Select Name from StudentMaster
    Where Name LIKE '_a %';
Example

- ‘computer%’ – matches any string begin with computer
- ‘%engg’ – matches any string containing ‘engg’ as substring
- ‘_s%’ – matches any string with second character ‘s’
- ‘_ _ _’ – matches any string with exactly three characters
- ‘_ _ _ %’ – matches any string of at least three character