ORACLE SQL

**SQL >>**

Stands For Structured Query Language With SQL We Can Store, Retrieve, Select, Extract, Read, Update, And Delete The Data In The Database.

**WHAT IS DATA?**

The World Is Divided Into Two Things One Is Non-Living ThingsAnother One Living Things But in Programing Language i.e. C, C++, JAVA, PYTHON, PHP Everything Will Be Consider as Object.

Data Describes the Properties of an Object

(Or)

Data Describes the Attributes of an Entity

 EXAMPLE -1

****OBJECT

 **PROPERTIES DATA**

|  |
| --- |
| **NAME NIBBA**  |
|  **AGE 24**  |
|  **GENDER MALE** |
|  **DATEOFBIRTH 14-FEB-2000** |
|  **HEIGHT 5’1’’ FEET** |
|  **WEIGHT 56KGS** |
|  **QUALIFICATION B.TECH** |
|  **INSTA\_ID call\_me\_nibba** **GIRL\_FRIEND\_NAME NIBBI** |

**HUMAN**

EXAMPLE:-2

**Properties or Attributes**

 ENTITY (OBJECT)

**WHAT IS DATABASE?**

Database Is Place (or) A Medium Which We Store the Data

In A Systematic and Organized Manner

 

The Basic Operations That Can Be Performed On Database Are

* CREATE / INSERT
* READ / RETRIEVE / SELECT / EXTRACT
* UPDATE / MODIFY
* DELETE / DROP

These Operations Are Referred As **CRUD** Operations.

 **DATABASE MANAGEMENT SYSTEM (DBMS)**:

"It is a software which is used to maintain and manage the database”

 **“Security** and **authorization”** are the two important features that DBMS provides.

 

|  |  |
| --- | --- |
|  | Software Data    Database |
| QUERY  LANGUAGE |  |
|  |
|  FILE FORMAT |  |

 DBMS

We use query language to communicate or interact with DBMS

 DBMS stores the data in the form of **files***.*

 **RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):**

 It is a type of DBMS software in which we store the data in the form of

 Tables (rows & columns) ".

 

 STRUCTURED QUERY

 LANGUAGE

Database

Data

*Software*

 TABLE (ROW&COLUMNS)

We use SQL to communicate or interact with RDBMS

 RDBMS stores the data in the form of **Tables.**

EXAMPLE

 **EMPLOYEE TABLE WITH DATA**

****

**RELATIONAL MODEL**

Relational Model was designed by “EDGER **FRANK CODD” (“TED”)**

 In Relational Model we can store the data in the form of *tables*

 Any DBMS which follows Relational Model becomes RDBMS.

DBMS.

Relational Model

RDBMS



**TABLE:**

"It is a logical organization of data which consists of Columns &Rows”

 (Or)

“The Combination of Horizontal Lines and Vertical lines Is Knows as Table.”

Rows

/

 Records

/

Tuples

Columns / Attribute / Fields

Cell

 Employee:

|  |  |  |
| --- | --- | --- |
| **EID**  | **ENAME** | **SALARY**  |
| 1 | SMITH | 1000 |
| 2 | ALLEN | 1500 |
| 3 | CLARK | 2000 |

RULES OF E.F CODD

Rule No:1

1. The data entered into a cell must always be a “**single valued data.”**

Example:

|  |  |  |
| --- | --- | --- |
| **EID**  | **ENAME** | **PHONE\_NO** |
| 1 | SMITH | 101 |
| 2 | ALLEN | 102 , 202  |
| 3 | CLARK | 103 |

|  |  |  |  |
| --- | --- | --- | --- |
| **EID**  | **ENAME** | **PHONE\_NO** | **ALTERNATE NO** |
| 1 | SMITH | 101 |  |
| 2  | ALLEN | 102 | 202 |
| 3 | CLARK | 103 |  |

Rule No 2

 In RDBMS we store everything in the form of tables including metadata

 Example: *Metadata:* The details about a data is knows as Metadata.

### MetaTable

PHOTO

DATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **EID**  | **ENAME** | **PHOTO**  |
|  | 1 | SMITH |  |  |  |  |
|  |
| 2 | ALLEN |  |
| 3 | CLARK |  |

Image Name : Mypic size : 127kb resolution : 400 x 600 format : jpeg

|  |  |  |  |
| --- | --- | --- | --- |
| **Image name**  | **size** | **Format**  | **Resolution**  |
| Mypic | 127 | jpeg | 400 x 600 |
|  |  |  |  |

 

 **Metadata is automatically generated (Auto generated)**

**Rule No: 3**

1. According to EDGER FRANK we can store the data in Multiple Tables If needed we can establish a connection between the tables with the Help of “**Key Attribute.”**

**Using (primary key, foreign key)**

****

 ****

**Rule No 4**

The data entered into the table can be validated in 2 steps.

* 1. By assigning Datatypes To column(s)
	2. By assigning Constraints To column(s)

Note point:-

**Datatypes are mandatory, Constraints are Optional.**

**DATATYPES**

 **It is used to specify or determine the type of data**

**That will be stored in a particular memory location.**

**Datatypes in SQL**

1. CHAR(SIZE) DATA TYPE
2. VARCHAR (size)/ VARCHAR2(size) DATA TYPE
3. DATE DATA TYPE
4. NUMBER(PRECISION ,[SCALE])
5. LARGE OBJECTS
	1. Character Large Object. (CLOB)
	2. Binary Large Object (BLOB)

**KEEP IN MIND!**

**NOTE: SQL is not a Case Sensitive Language.**

**CHAR (SIZE) DATA TYPE;-**

1. In character datatype we can store 'A-Z' , 'a-z' , '0-9' And Special Characters( $ , & , @ , ! … ) .
	* Characters must always be enclosed within single quotes ‘ASHOKIT '.(‘ ‘)
	* Whenever we use char datatype we must mention size
	* **Size:** it is used to specify number of characters it can store.
		+ The maximum number of characters it can store is

**2000ch**.

Char follows **FIXED LENGTH MEMORY ALLOCATION**

Syntax: CHAR (SIZE)

Example: CHAR (8)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **R** | **A** | **M** |  |  |  |  |  |

Used Memory Unused Memory (memory wastage)

 In character datatype we can store ALPHANUMARIC

(COMBINATION OF ALPHA BITES AND NUMARIC TYPE OF DATE)

 EXAMPLE ‘ABC123’

**VARCHAR (SIZE) DATA TYPE:-**

1. In varchar datatype we can store 'A-Z' , 'a-z' , '0-9' And Special Characters( $ , & , @ , ! … ) .

Characters must always be enclosed within single quotes ' RAMAKRISHNA SIR'.

Whenever we use char datatype we must mention size

**Size**: it is used to specify number of characters it can store.

 The maximum number of characters it can store **2000 CHARACTERS**

**Varchar (size) follows “variable length memory allocation** “.

Syntax: VARCHAR (SIZE)

Example: VARCHAR (8)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **R** | **A** | **M** |  |  |  |  |  |

 Used Memory Unused Memory (free memory)

* In character datatype we can store ALPHANUMARIC

(COMBINATION OF ALPHA BITES AND NUMARIC TYPE OF DATE)

 EXAMPLE: Insta\_Id ‘link2ram’

**VARCHAR2 (SIZE) DATA TYPE:-**

In varchar2 datatype we can store 'A-Z', 'a-z' , '0-9' And Special

Characters( $ , & , @ , ! … ) .

Characters must always be enclosed within single quotes ' RAMAKRISHNA SIR'.

Whenever we use char datatype we must mention size

**Size**: it is used to specify number of characters it can store.

The maximum number of characters it can store **4000 CHARACTERS**

**Varchar2 (size) follows “variable length memory allocation** “.

Example: VARCHAR (8)

 Syntax: VARCHAR2 (SIZE)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **R** | **A** | **M** |  |  |  |  |  |

 Used Memory Unused Memory (free memory)

* In character datatype we can store ALPHANUMARIC

(COMBINATION OF ALPHA BITES AND NUMARIC TYPE OF DATE)

 EXAMPLE: Insta\_Id ‘xyz123’

**Example:**

By using char (size) Data Type Varchar2 (size**)**

## STUDENT

|  |  |  |  |
| --- | --- | --- | --- |
| **USN** | **SNAME** | **ADDRESS** | **PAN\_NO** |
| CHAR(4) | VARCHAR(10) | VARCHAR2(10) | CHAR(10) |
| RAM | NIBBA | AMEERPET | ABC123XYZ1 |
| RAVI | NIBBI | KPHB | ABC123XYZ2 |

**ASSIGNMENT:**

DIFFERENTIATE BETWEEN CHAR & VARCHA

 **ASCII: [American Standard Code for Information Interchange**]

 **SELECT ASCII (‘A’)**

 **FROM DUAL;**

|  |  |
| --- | --- |
| 'A' | 65 |
| 'Z' | 90 |
| 'a' | 97 |
| 'z' | 122 |

**NUMBER (PRECISION,[SCALE])**

SYNTAX: **NUMBER** (Precision, **[Scale**])

**NUMBER:** It is used to store numeric values.

 [ ] - Not Mandatory.

**Precision**: it is used to determine the number of digits used to store integer value

**Scale**: it is used to determine the number of digits used to store decimal (floating) value within the precision

 By default scale value is zero (**0**)

 Scale is not mandatory

 EXAMPLE WE HAVE TAKEN DIGIT 7

|  |  |  |
| --- | --- | --- |
| Example : | Number ( 3 ) | +/- 777 |
| Example : | Number ( 5 , 0 ) | +/- 77777 |
| Example : | Number ( 5 , 2 ) | +/- 777.77 |
| Example : | Number ( 7 , 3 ) | +/- 7777.777 |
| Example : | Number ( 4 , 4 ) | +/- .7777 |
| Example : | Number ( 5 , 4 ) | +/- 7.7777 |
| Example : | Number ( 3 , 6 ) | +/- .000777 |
| Example : | Number ( 5 , 8 ) | +/- .00077777 |
| Example : | Number ( 2 , 7 ) | +/- .0000077 |

|  |  |  |
| --- | --- | --- |
| **EID** | **PHONE\_NO** | **SALARY** |
| Number( 3 ) | Number ( 10 ) | Number ( 7 , 2 ) |
| 101 | 9985396677 |  25000.65 |

 **DATE:** it is used to store dates in a particular format.

 It used Oracle specified Format.

|  |  |  |
| --- | --- | --- |
| 'DD-MON-YY' | OR | 'DD-MON-YYYY' |
| '12-AUG-24' |  | '12-AUG-2024' |

SYNTAX: **DATE**

 **EMPLOYEE\_DETAILS**

|  |  |  |
| --- | --- | --- |
| **EMPLOYE\_DOB** | **EMPLOYEE\_HIREDATE** | **EMPLOYEE\_ANNIVERSARY** |
| DATE | DATE | DATE |
| ’22-JUN-2000’ | ’22-MAY-2018’ | ’24-MAY-2024’ |

 **LARGE OBJECTS**

 **CHARACTER LARGE OBJECT (CLOB)**

Datatype we can store 'A-Z' , 'a-z' , '0-9' And Special Characters( $ , & , @ , ! … ) .

It is used to store characters up to 4 GB of size.

In character datatype we can store ALPHANUMARIC.

(COMBINATION OF ALPHA BITES AND NUMARIC TYPE OF DATE)

CHARACTER LARGE OBJECT VALUES WILL BE STORED IN CHARACTER

FORMAT

|  |
| --- |
| SYNTAX : **CLOB** |

 EXAMPLE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| @ | # | $ | ^ | 4 | 5 |
| E | 4 | 7 | 8 | % | \* |
| 5 | ^ | ( | \_\* | # | $ |

**Binary large object (BLOB):**

Datatype we can store 'A-Z, 'a-z', '0-9' And Special Characters ($, &, @, #…).

It is used to store characters up to 4 GB of size.

In Binary Large Object Datatype we can store the Values/data in Binary Format

|  |
| --- |
| SYNTAX: **BLOB** |

 EXAMPLE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1` | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |