**DATA INTEGRITY**

**DATA INTEGRITY:**

Data integrity is **used to restrict invalid data into the table.**

We can **achieve data integrity** by

1. Data type
2. Constraints

**EX:**

|  |  |
| --- | --- |
| Number | Name |
| -3446 | Mahe |

**1.DATA TYPE:**

1. It is nothing but the **type of data given to every individual column.**

**TYPES OF DATA TYPE:**

1. NUMBER
2. CHAR/VARCHAR
3. DATE.

**NUMBER:**

1. It is used to **input the numeric value into a table column.**

EX:

1. Number(2) 🡪 -99 to 99
2. Number(4) 🡪 -9999 to 9999
3. Number(4,2) 🡪 -99.99 to 99.99
4. Number(8,2) 🡪 -999999.99 to 999999.99

**CHAR/VARCHAR:**

|  |  |
| --- | --- |
| **CHAR** | **VARCHAR** |
| Here we can **enter up to 2000 characters.** | Here we can **enter 4000 characters.** |
| It will **allow null values.** | It will **not allow null values.** |
| It has **fixed length.** | It has **variable length.** |
| It **allows only characters.** | It **allows alphanumeric.** |
| Memory consumption. | **No memory consumption.** |

**DATE:**

* Here we have default method called as **DATE to use to enter** the **different type date format.**

**default format :-** dd – mon – yy

**BLOB:**

Stands for – Binary Large Object

It stores binary data (images, movies, music files) within the database. It stores upto 4GB.

**CLOB:**

Stands for – Character Large Object

It stores plain character data like **varchar** field upto 4GB.

SQL Data Types

The SQL data type defines a kind of value that a column can contain.

In a database table, every column is required to have a name and a data type.

**Data Type varies from database to database. For example, MySQL supports INT but Oracle supports NUMBER for integer values.**

These are the general data types in SQL.

|  |  |  |
| --- | --- | --- |
| **Data-type** | **Syntax** | **Explanation** |
| Integer | INTEGER | The integer data type is used to specify an integer value. |
| Smallint | SMALLINT | The smallint data type is used to specify small integer value. |
| Numeric | NUMERIC(P,S) | It specifies a numeric value. Here 'p' is precision value and 's' is scale value. |
| Real | REAL | The real integer is used to specify a single precision floating point number. |
| Decimal | DECIMAL(P,S) | It specifies a decimal value. Here 'p' is precision value and 's' is scale value. |
| Double precision | DOUBLE PRECISION | It specifies double precision floating point number. |
| Float | FLOAT(P) | It specifies floating-point value e.g. 12.3, 4.5 etc. Here, 'p' is precision value. |
| Character | CHAR(X) | Here, 'x' is the character's number to store. |
| Character varying | VARCHAR2(X) | Here, 'x' is the character's number to store |
| Bit | BIT(X) | Here, 'x' is the number of bits to store |
| Bit varying | BIT VARYING(X) | Here, 'x' is the number of bits to store (length can vary up to x). |
| Date | DATE | It stores year, month and days values. |
| Time | TIME | It stores hour, minute and second values |
| Timestamp | TIMESTAMP | The timestamp data type is used to store year, month, day, hour, minute and second values. |
| Time with time zone | TIME WITH TIME ZONE | It is exactly same as time but also store an offset from UTC of the time specified. |
| Timestamp with time zone | TIMESTAMP with TIME ZONE | It is same as timestamp but also stores an offset from UTC of the time specified. |

**2.Constraints:**

1. Constraints are nothing but **condition** which **restricts invalid data** in the tables.
2. It is **provided to the columns** of the table.

**Types of constraints:**

1. Not Null
2. Unique
3. Check
4. Primary key
5. Foreign key

**NULL:**

1. NULL is nothing but it is **neither zero nor blank space.**
2. Null **doesn’t occupy any space** in the memory.
3. Null represent **unknown value.**
4. **Two null never ever occur** in oracle.
5. If we **perform** **any operation** (arithmetic operation) with **null** its **result** in **null** itself.

**Ex:**

* Null+100=null
* Null\*109=null

**NOT NULL:**

1. Not null value constraint ensures **some value present** in the **column**.

**UNIQUE:**

1. It **won’t accept the duplicate value** in column.
2. It can **take multiple null values.**

**CHECK:**

1. It is nothing but we are **providing additional validation** as per the **customer requirement specification.**

**Ex:**

Check(Phoneno=10)

**PRIMARY KEY:**

1. It is the combination of **not null** and **unique** constraint.
2. It is column(s) which **uniquely identifies** a row.
3. Creation of primary key is **not mandatory** but **highly recommend**.
4. In a table **only one** primary key is allowed.

**EX:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Empno | Ename | Sal | Dept | Hiredate |
| 1 | Mahe | 50000 | Dev | 18-7-1995 |
| 2 | Gokul | 55000 | Dev | 18-7-2017 |
| 3 | Viky | 20000 | Test | 15-3-2017 |
| 4 | Vicky | 30000 | Test | 16-3-2012 |

**CANDIDATE KEY:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Empno | Ename | Sal | Dept | Hiredate | Phoneno | Email id |
| 1 | Mahe | 50000 | Dev | 18-7-1995 | 9148848975 | [Maherms95@gmail.com](mailto:Maherms95@gmail.com) |
| 2 | Gokul | 55000 | Dev | 18-7-2017 | 9312534642 | gokulkala@gmail.com |
| 3 | Viky | 20000 | Test | 15-3-2017 | 9346312423 | Vicky@gmail.com |
| 4 | Vicky | 30000 | Test | 16-3-2012 | 9325263673 | Vicky213@gmail.com |

1. **Columns** which are **eligible to become primary key** is called as candidate key.

* Not Null+unique = candidate key (primary key is allowed)

**ALTERNATE KEY:**

1. **Column** which are **eligible to become primary key but not chosen as primary key.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Empno | Ename | Sal | Dept | Hiredate | phoneno | Email id |
| 1 | Mahe | 50000 | Dev | 18-7-1995 | 9148848975 | [Maherms95@gmail.com](mailto:Maherms95@gmail.com) |
| 2 | Gokul | 55000 | Dev | 18-7-2017 | 9312534642 | gokulkala@gmail.com |
| 3 | Viky | 20000 | Test | 15-3-2017 | 9346312423 | Vicky@gmail.com |
| 4 | Vicky | 30000 | Test | 16-3-2012 | 9325263673 | Vicky213@gmail.com |

NOT NULL+unique = alternate key (primary key is not allowed)

**FOREIGN KEY:**

1. Foreign key is **use to create relationship between two table.**
2. It is also called as **referential integrity constraint.**
3. In a table, we can **have more than one foreign key.**
4. **Foreign key** is **created in child table.**
5. **Foreign key of child table** will be the **primary key of master table.**
6. It can **accept null and duplicate value.**

**EX:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Empno | Ename | Sal | Comm | Hiredate | Deptno | dname | Dloc |
| 1 | Chandru | 10000 |  | 15-mar-2017 | 10 | Hr | Chennai |
| 2 | Raja | 20000 | 0 | 5-may-2013 | 30 | Dev | Ban |
| 3 | Maha | 23000 |  | 8-aug-2015 | 20 | Test | Cbe |
| 4 | Raji | 13000 | 500 | 9-feb-2016 | 20 | Dev | Chennai |
| 5 | Raja | 54000 | 1000 | 5-jan-2011 | 10 | Test | Ban |
| 6 | Golu | 65342 |  | 9-apr-2016 | 30 | Hr | Cbe |

**EX:Employee table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Empno | Ename | Sal | Comm | Hiredate | Deptno | Project id |
| 1 | Chandru | 10000 |  | 15-mar-2017 | 10 | 101 |
| 2 | Raja | 20000 | 0 | 5-may-2013 | 30 | 302 |
| 3 | Maha | 23000 |  | 8-aug-2015 | 20 | 203 |
| 4 | Raji | 13000 | 500 | 9-feb-2016 | 20 | 202 |
| 5 | Raja | 54000 | 1000 | 5-jan-2011 | 10 | 104 |
| 6 | Golu | 65342 |  | 9-apr-2016 | 30 | 306 |

**EX: dept table**

|  |  |  |
| --- | --- | --- |
| Deptno | dname | Dloc |
| 10 | Hr | Chennai |
| 30 | Dev | Ban |
| 20 | Test | Cbe |
| 20 | Dev | Chennai |
| 10 | Test | Ban |
| 30 | Hr | Cbe |

**EX: project table**

|  |  |  |  |
| --- | --- | --- | --- |
| Deptno | dname | Dloc | Project id |
| 10 | Hr | Chennai | 101 |
| 30 | Dev | Ban | 302 |
| 20 | Test | Cbe | 203 |
| 20 | Dev | Chennai | 202 |
| 10 | Test | Ban | 104 |
| 30 | Hr | Cbe | 306 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | **Check (sal > 0)** |  |
|  |  |  |  |  |  |  |  |  |  |
| **PK** | **NN** |  |  | **PK** | **NN** | **FK** | **NN** | **Unique** | **Unique** |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Dept No.** | **Dept name** |  |  | **Emp No.** | **Emp Name** | **Dept No.** | **Salary** | **Ph No.** | **Email** |
| 10 | Accounting |  |  | 101 | A | 10 | 200000 | 2222 | [a@gmail](mailto:a@gmail) |
|  |  |  |  |  |  |  |  |  |  |
| 20 | Research |  |  | 102 | B | 10 | 30000 | - | - |
|  |  |  |  |  |  |  |  |  |  |
| 30 | Sales |  |  | 103 | C | 20 | 400000 | 3333 | - |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **RELATIONSHIP** | | | | | | |  | **NULL** | |

**EX FOR CONSTRAINT:**