**Software Engineering Assignment**

**MODULE: 5 (Database)**

**Basics of Database**

1. What do you understand by Database?

Ans. Database is a collection of information or data. It is possible to store, retrieve and edit this data by software called DBMS (**D**ata**b**ase **M**anagement **S**ystem).

1. What is Normalization?

Ans. Normalization is the process of minimizing redundancy (duplicity) from the database table. It divides larger table into smaller and links them using relationships. It works through series of stages called Normal Forms. Following are the various types of Normal Forms:

1. **First Normal Form (1NF):** It helps to eliminate duplicate data and simply queries.
2. **Second Normal Form (2NF):** It divides data into subgroups that exist in multiple rows of database table and represents it in a new table with connections made between them.
3. **Third Normal Form (3NF):** The relation that is in 1NF and 2NF, where none of the non-primary key attributes transitively depend on their primary keys.
4. **BCNF (Boyce Codd Normal Form):** It is an advanced version of 3NF.
5. What is difference between DBMS and RDBMS?

|  |  |  |
| --- | --- | --- |
|  | **DBMS** | **RDBMS** |
| 1. | DBMS stands for Database Management System | RDBMS stands for Relational Database Management System |
| 2. | Data is stored in form of files | Data is stored in form of tables |
| 3. | There is no existence of relationship between data | The data stored is related to each other |
| 4. | Normalization is not present | Normalization is present |
| 5. | It deals with small quantity of data | It stores large amount of data |
| 6. | Data redundancy is common in this model | It does not allow data redundancy |
| 7. | Security is less | More security measures are provided |
| 8. | It supports single user | It supports multiple user |

1. What is MF Codd Rule of RDBMS?

Ans. Dr. Edgar F. Codd (E.F Codd) was a Computer Scientist who, in 1985, invented the Relational model for Database management(RDBMS). He proposed 13(numbered 0-12) rules popularly known as Codd's 12 rules that actually defines what quality a DBMS requires in order to become a Relational Database Management System(RDBMS).

He stated the following rules:

Rule 0: The Foundation Rule

Rule 1: Information Rule

Rule 2: Guaranteed Access Rule

Rule 3: Systematic Treatment of Null Values

Rule 4: Active/Dynamic Online Catalog based on the relational model

Rule 5: Comprehensive Data SubLanguage Rule

Rule 6: View Updating Rule

Rule 7: Relational Level Operation (High-Level Insert, Update and Delete) Rule

Rule 8: Physical Data Independence Rule

Rule 9: Logical Data Independence Rule

Rule 10: Integrity Independence Rule

Rule 11: Distribution Independence Rule

Rule 12: Non Subversion Rule

1. What do you understand by Data Redundancy?

Ans. Data Redundancy means when multiple copies of same information (data) is stored in more than one place at a time.

1. What is DDL Interpreter?

Ans. It interprets the DDL (Data Definition Language) instructions (tabular format) and stores the record in data dictionary/metadata (data of the database).

1. What is DML compiler in SQL?

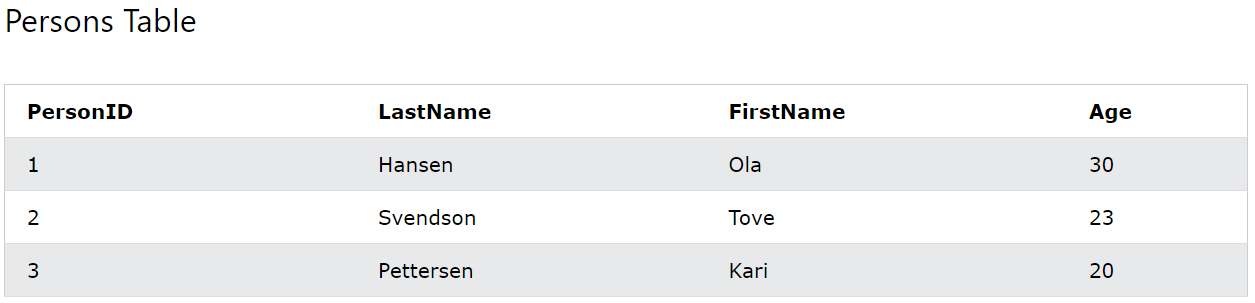
Ans. It converts DML (Data Manipulation Language) instructions into machine language (low level language).

1. What is SQL Key Constraints writing an example of SQL key constraints?

Ans. SQL Key Constraints are used to specify rules for the data in a table. It limits the type of data that goes into the table to ensure the accuracy and reliability of the data in the table. Any violation to this will abort the action.

The following constraints are commonly used in SQL:

1. **Primary Key**: It is a combination of not null (never blank) and unique data.

Example:

CREATE TABLE Persons (

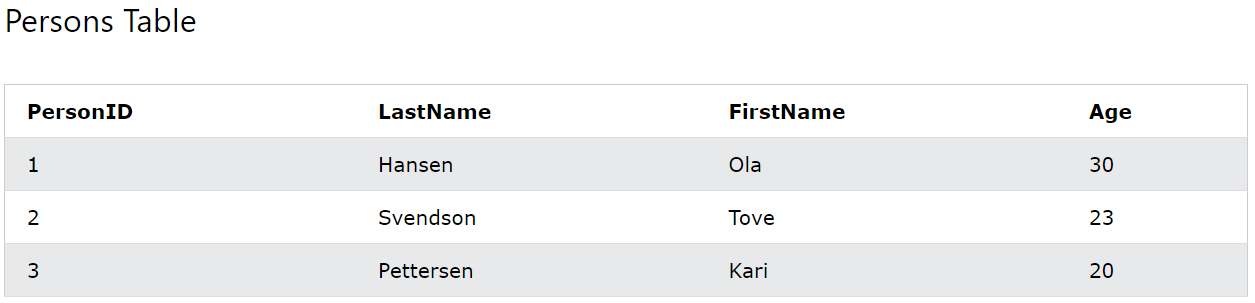
**ID int NOT NULL PRIMARY KEY,**

LastName varchar(50) NOT NULL,

FirstName varchar(50),

Age int );

1. **Unique Key**: It ensures that data is not repeated (values in every column is different) and can be null.

Example:

CREATE TABLE Persons (

**ID int NOT NULL UNIQUE,**

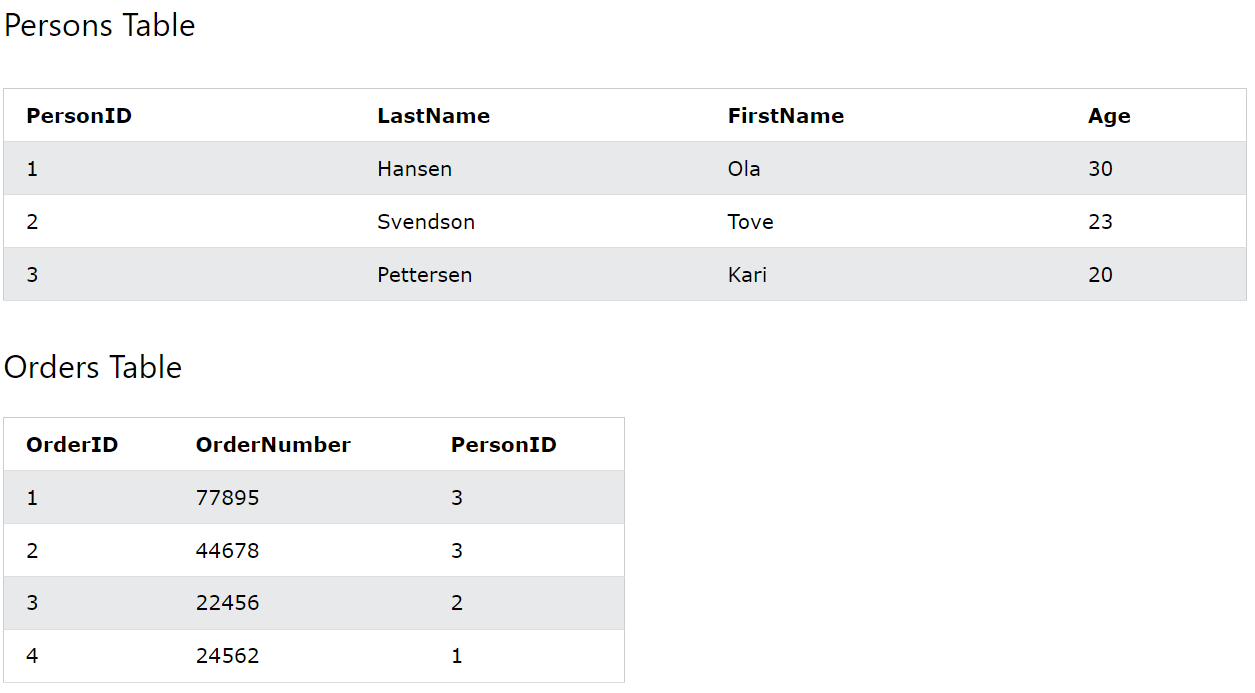
LastName varchar(50) NOT NULL,

FirstName varchar(50),

Age int );

1. **Foreign Key:** It is used to maintain a relation between two tables.

Example:



CREATE TABLE Persons (

**ID int NOT NULL PRIMARY KEY,**

LastName varchar(50) NOT NULL,

FirstName varchar(50),

Age int

);

CREATE TABLE Orders (

OrderID int NOT NULL,

OrderNumber int NOT NULL,

PersonID int,

PRIMARY KEY (OrderID),

**FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)**

);

1. What is Save Point? How to create a save point? Write a query?

Ans. Save Point is a command in SQL to roll back to a certain point without rolling back the entire table.

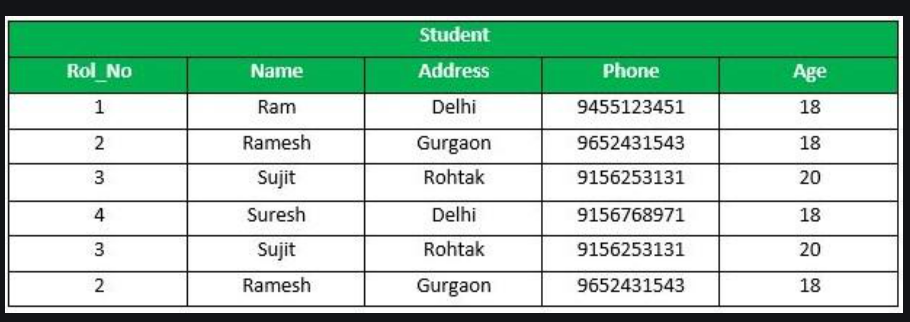
* Syntax for Savepoint command:

SAVEPOINT SAVEPOINT\_NAME;

* Syntax for rolling back to Savepoint command:

ROLLBACK TO SAVEPOINT\_NAME;

* **Query Example:**

*From the above example Sample table1, Delete those records from the table which have age = 20 and then ROLLBACK the changes in the database by keeping Savepoints.*

**Query:**

SAVEPOINT SP1; //Savepoint created

DELETE FROM Student WHERE Age=20; //deleted

SAVEPOINT SP2; //Savepoint created

**Output:**

1. What is trigger and how to create a trigger in SQL?

Ans. Trigger is used to store history of the new data inserted/updated/deleted in predefined database. It automatically updates a record in one table whenever a record is inserted into another table.

Syntax to create a trigger:

DELIMITER $$

CREATE TRIGGER `TRIGGER\_NAME` (AFTER | BEFORE) {INSERT | UPDATE | DELETE} ON `table\_name` FOR EACH ROW

BEGIN

INSERT into `table\_name` SET `column\_name = new.column\_name | old.column\_name;

END$$