TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# [Module 5] Database

## DBMS

- DBMS stands for Data Base Management System.
- Data + Management System
- Database is a collection of inter-related data and Management System is a set of programs to store and retrieve those data.
- DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.
- For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

## DBMS

Here is a list of some popular database.

- MySQL
- Microsoft Access
- Oracle
- PostgreSQL
- dBASE
- FoxPro
- SQLite
- IBM DB2
- LibreOffice Base

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# Need of DBMS

- Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data
  Storage:
- According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage.
- Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## RDBMS

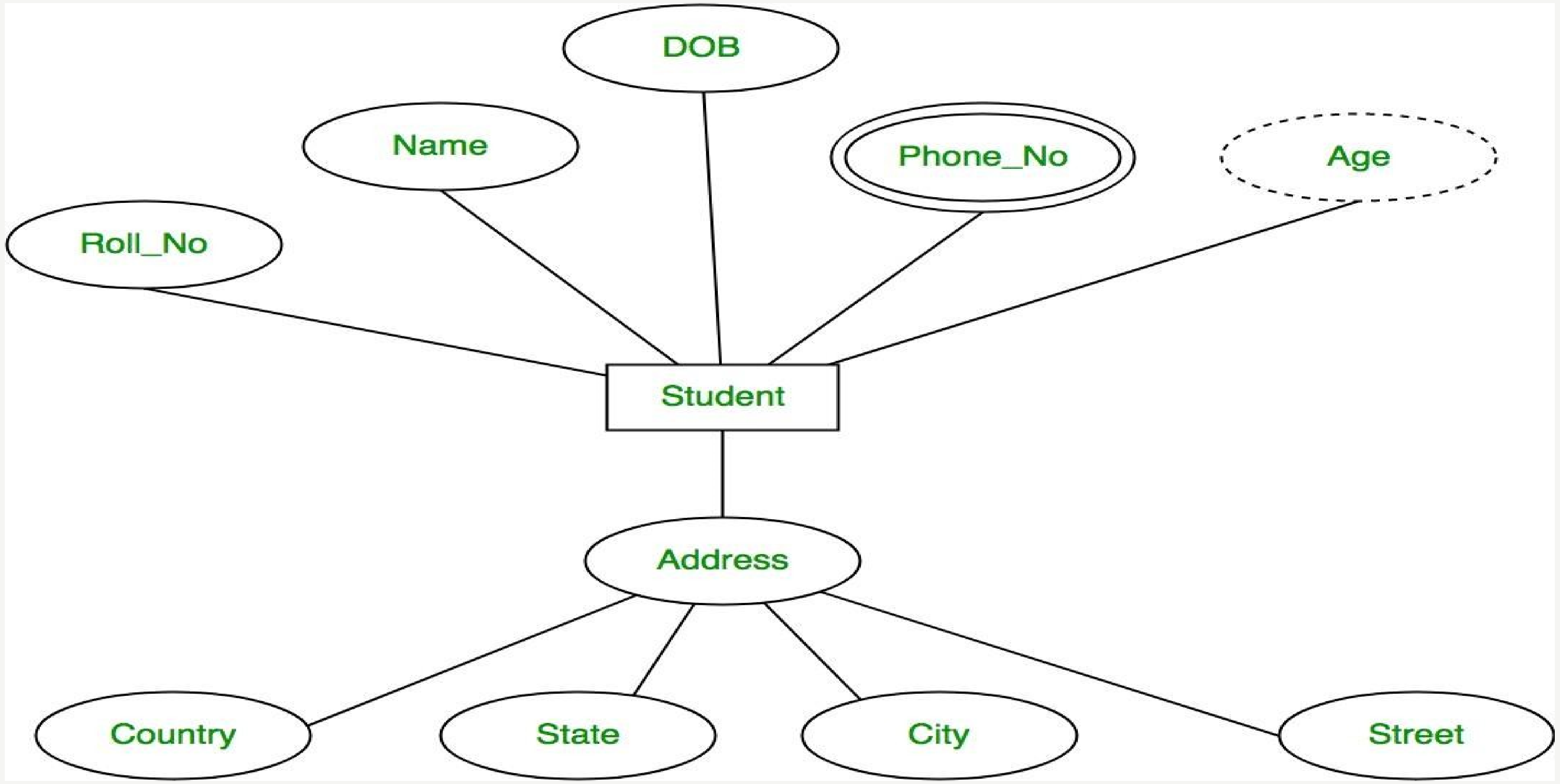
- Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data
  Storage:
- According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage.
- Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

# E-R Model

- The ER or (Entity Relational Model) is a high-level conceptual data model diagram.
- Entity-Relation model is based on the notion of real-world entities and the relationship between them.
- ER modeling helps you to analyze data requirements systematically to produce a well-designed database.
- So, it is considered a best practice to complete ER modeling before implementing your database.
- Entity relationship diagram displays the relationships of entity set stored in a database.
- In other words, we can say that ER diagrams help you to explain the logical structure of databases.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# E-R Model



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# Algebra

- Relational Algebra is procedural query language, which takes Relation as input and generate relation as output. Relational algebra mainly provides theoretical foundation for relational databases and SQL.

**Unary Relational Operations**
- SELECT (symbol: σ)
- PROJECT (symbol: π) RENAME (symbol: )

**Relational Algebra Operations From Set Theory**
- UNION (υ) INTERSECTION ( ), DIFFERENCE (-) CARTESIAN PRODUCT ( x )

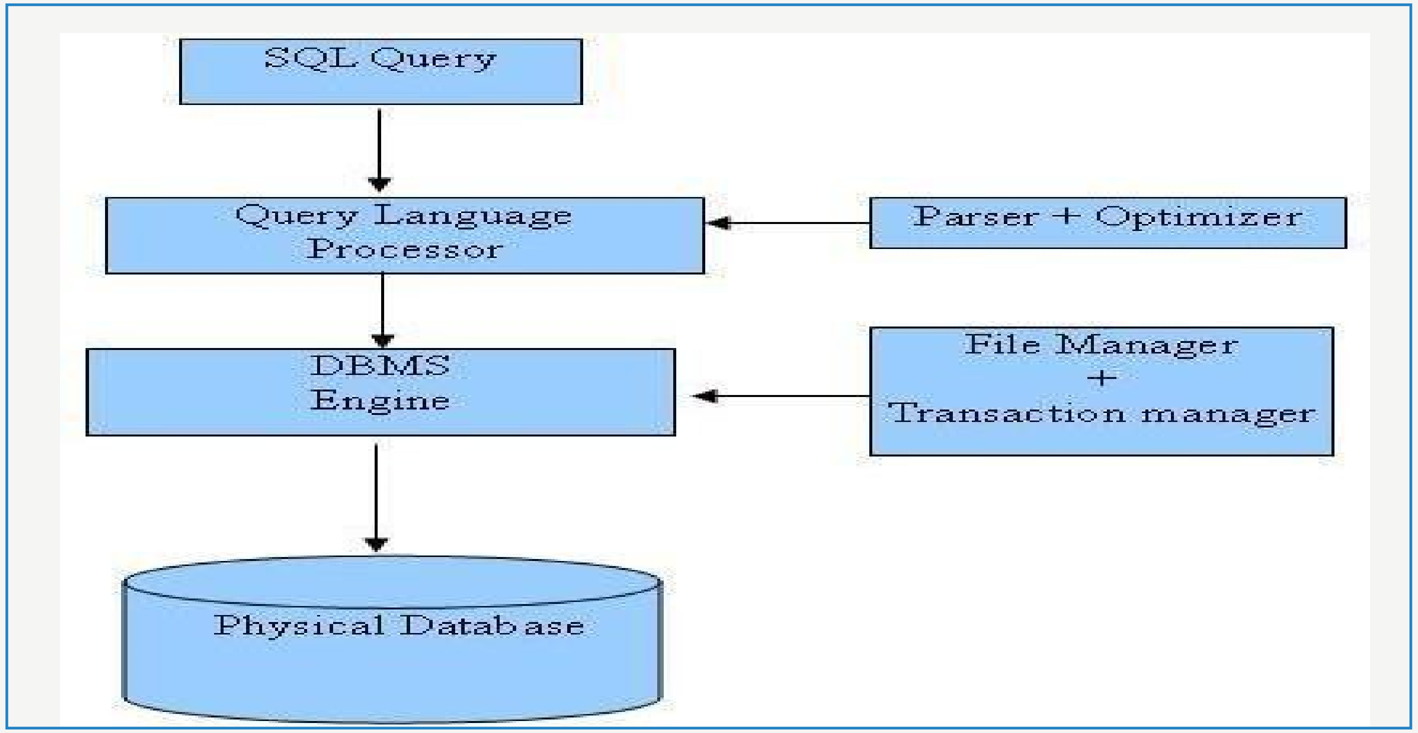
**Binary Relational Operations**
- JOIN DIVISION

## What is SQL?

- SQL is Structured Query Language, which is a computer language for storing,
- manipulating and retrieving data stored in relational database.
- SQL is a language of database, it includes database creation, deletion, fetching rows and modifying rows etc.
- SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server use SQL as standard database language.
- Also, they are using different dialects, such as:
- MS SQL Server using T-SQL, ANSI SQL
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc

# Objectives

- "The large majority of today's business applications revolve around relational databases and the SQL programming language (Structured Query Language). Few businesses could function without these technologies…"



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## What is SQL?

- Allows users to access data in relational database management systems. Allows users to describe the data.
- Allows users to define the data in database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database. Allows users to set permissions on tables, procedures, and views
- SQL stands for Structured Query Language
- SQL allows you to access a database
- SQL is an ANSI standard computer language
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert new records in a database

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## What is SQL?

- SQL can delete records from a database
- SQL can update records in a database
- SQL is easy to learn
- action queries insert, update & delete data
- select queries retrieve data from DB

## Constraints(Keys)

**Primary Key:**

- A primary key is a column of table which uniquely identifies each
- tuple (row) in that table.
- Primary key enforces integrity constraints to the table.
- Only one primary key is allowed to use in a table.
- The primary key does not accept the any duplicate and NULL values.
- The primary key value in a table changes very rarely so it is chosen with care
- where the changes can occur in a seldom manner.
- A primary key of one table can be referenced by foreign key of another table.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## Constraints(Keys)

**Unique Key:**

- Unique key constraints also identifies an individual table uniquely in a relation or table.
- A table can have more than one unique key unlike primary key.
- Unique key constraints can accept only one NULL value for column.
- Unique constraints are also referenced by the foreign key of another table.

# Constraints(Keys)

## Foreign Key:

- When, "one" table's primary key field is added to a related "many" table in order to create the common field which relates the two tables, it is called a foreign key in the "many" table.
- In the example given below, salary of an employee is stored in salary table.
- Relation is established via is stored in "Employee" table. To identify the salary of "Jforeign key column "Employee_ID_Ref" which refers "Employee_ID" field in Employee table.
- of "Jhon" is stored in "Salary" table. But his employee info
- For example, salary hon", his "employee id" is stored with each salary record.

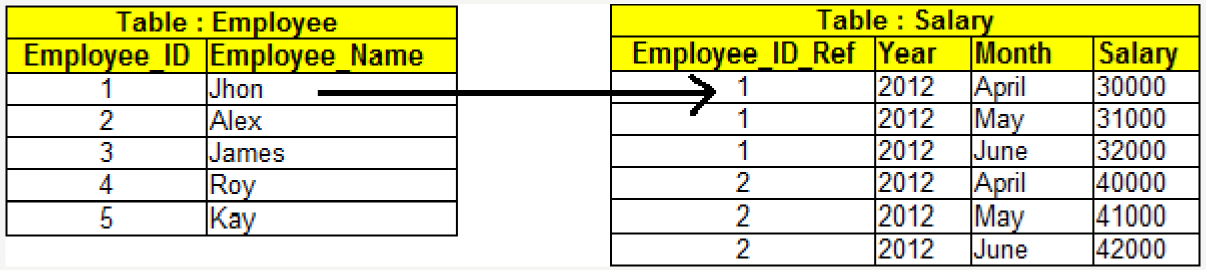


Table : Employee

| Employee_ID | Employee_Name |
|---|---|
| 1 | Jhon |
| 2 | Alex |
| 3 | James |
| 4 | Roy |
| 5 | Kay |

Table : Salary

| Employee_ID_Ref | Year | Month | Salary |
|---|---|---|---|
| 1 | 2012 | April | 30000 |
| 1 | 2012 | May | 31000 |
| 1 | 2012 | June | 32000 |
| 2 | 2012 | April | 40000 |
| 2 | 2012 | May | 41000 |
| 2 | 2012 | June | 42000 |

# Database Normalization

- Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations.
- Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.
- Most Commonly used normal forms:

## First Normal Form:

- First normal form(1NF) Second normal form(2NF) Third normal form(3NF) Boyce & Code normal form (BCNF)
- If a relation contain composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute.
- A relation is in first normal form if every attribute in that relation is singled valued attribute.

# Database Normalization

**Second Normal Form:**

- To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency.
- relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.
- Partial Dependency – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

# Database Normalization

**Third Normal Form:**

- A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.
- A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y

  X is a super key.

  Y is a prime attribute (each element of Y is part of some candidate key).
- Transitive dependency – If A->B and B->C are two FDs then A->C is called transitive dependency.

## SQL Statement Types

- DDL – Data Definition Language
- DML – Data Manipulation Language
- DCL – Data Control Language
- DQL – Data Query Language
- **SQL Join Types**
- INNER JOIN: returns rows when there is a match in both tables.
- LEFT JOIN: returns all rows from the left table, even if there are no
- matches in the right table.
- RIGHT JOIN: returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN: returns rows when there is a match in one of the tables.

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# SQL Statement Types

## DDL - Data Definition Language

| Command | Description |
|---|---|
| CREATE | Creates a new table, a view of a table, or other object in database |
| ALTER | Modifies an existing database object, such as a table. |
| DROP | Deletes an entire table, a view of a table or other object in the database. |

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# SQL Statement Types

## DQL – Data Query Language

| Command | Description |
|---|---|
| SELECT | Retrieves certain records from one or more tables |

- **DML – Data Manipulation Language**

| Command | Description |
|---|---|
| INSERT | Creates a record |
| UPDATE | Modifies records |
| DELETE | Deletes records |

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# SQL Statement Types

## DCL – Data Control Language

| Command | Description |
|---|---|
| GRANT | Gives a privilege to user |
| REVOKE | Takes back privileges granted from user |

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## SQL Statement Types

**SQL CREATE DATABASE STATEMENT**
CREATE DATABASE database_name;

**SQL DROP DATABASE Statement:**
DROP DATABASE database_name;

**SQL USE STATEMENT**
USE DATABASE database_name;

**SQL CREATE TABLE STATEMENT**
CREATE TABLE table_name( column1 datatype, column2 datatype, column3 datatype, ..... , columnN datatype, PRIMARY KEY( one or more columns ) );

## SQL Statement Types

**SQL DROP TABLE STATEMENT**

DROP TABLE table_name;

**SQL TRUNCATE TABLE STATEMENT**

TRUNCATE TABLE table_name;

**SQL ALTER TABLE STATEMENT (RENAME)**

ALTER TABLE table_name RENAME TO new_table_name;

**SQL INSERT INTO STATEMENT**

INSERT INTO table_name( column1, column2....columnN) VALUES ( value1, value2.
valueN);

## SQL Statement Types

**SQL UPDATE STATEMENT**

UPDATE table_name SET column1 = value1, column2 = value2. columnN=valueN
[ WHERE CONDITION ];

**SQL DELETE STATEMENT**

DELETE FROM table_name WHERE {CONDITION}

**SQL SELECT STATEMENT**

SELECT column1, column2....columnN FROM table_name;

**SQL DISTINCT CLAUSE**

SELECT DISTINCT column1, column2....columnN FROM table_name;

## SQL Statement Types

**SQL WHERE CLAUSE**

SELECT column1, column2....columnN FROM table_name WHERE CONDITION;

**SQL AND/OR CLAUSE**

SELECT column1, column2....columnN FROM table_name WHERE CONDITION-1 {AND|OR} CONDITION-2;

**SQL IN CLAUSE**

SELECT column1, column2. column table_name WHERE column_name IN FROM (val-1, val-2, val-N);

**SQL BETWEEN CLAUSE**

SELECT column1, column2. column table_name WHERE column_name FROM BETWEEN val-1 AND val-2;

# SQL Statement Types

**SQL LIKE CLAUSE**

SELECT column1, column2.        column table_name WHERE column_name LIKE {
FROM PATTERN };

**SQL ORDER BY CLAUSE**

SELECT column1, column2.        columnN  table_name WHERE CONDITION ORDER
FROM BY column_name {ASC|DESC};

**SQL GROUP BY CLAUSE**

SELECT SUM(column_name) table_name WHERE CONDITION GROUP BY
FROM column_name;

**SQL COUNT CLAUSE**

SELECT COUNT(column_name)FROM table_name WHERE CONDITION;

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# SQL Statement Types

**SQL HAVING CLAUSE**
SELECT SUM(column_name) FROM table_name WHERE CONDITION GROUP BY column_name HAVING (arithematicfunction condition)

**SQL CREATE INDEX Statement :**
CREATE UNIQUE INDEX index_name ON table_name( column1, column2,...columnN);

**SQL DROP INDEX STATEMENT**
ALTER TABLE table_name DROP INDEX index_name;

**SQL DESC Statement :**
DESC table_name;

**SQL COMMIT STATEMENT**
COMMIT

# SQL Statement Types

**SQL ROLLBACK STATEMENT**

ROLLBACK;

## JOIN

A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them.

Different types of Joins are:

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. FULL JOIN

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# SQL Statement Types

- **1. Inner Join**
  The most frequently used and important of the joins is the INNER JOIN. They are also referred to as an EQUIJOIN.
- The INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. T
- he query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

**Syntax :**

```
SELECT table1.column1, table2.column2...FROM table1INNER JOIN table2ON
table1.common_filed = table2.common_field
```

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# SQL Statement Types

- **Left Join**

- The SQL LEFT JOIN returns all rows from the left table, even if there are no matches in the right table. This means that if the ON clause matches 0 (zero) records in right table, the join will still return a row in the result, but with NULL in each column from right table.
- This means that a left join returns all the values from the left table, plus matched values from the right table or NULL in case of no matching join predicate.
- **Syntax:**
- SELECT table1.column1, table2.column2...FROM table1LEFT JOIN table2ON table1.common_filed = table2.common_field;

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## SQL Statement Types

- **Right Join**
- The SQL RIGHT JOIN returns all rows from the right table, even if there are no matches in the left table. This means that if the ON clause matches 0 (zero) records in left table, the join will still return a row in the result, but with NULL in each column from left table.
- This means that a right join returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.

- **Syntax:**
- SELECT table1.column1, table2.column2...FROM table1RIGHT JOIN table2ON table1.common_filed = table2.common_field

# SQL Statement Types

- **Full Join**
- The SQL FULL JOIN combines the results of both left and right outer joins.
- The joined table will contain all records from both tables, and fill in NULLs for missing matches on either side.

- **Syntax:**
- SELECT table1.column1, table2.column2...FROM table1FULL JOIN table2ON table1.common_filed = table2.common_field;

# SQL Statement Types

## Function

- SQL has many built-in functions for performing calculations on
- data. They are divided into 2 categories:

1. Aggregate Function
2. Scalar Function

1. **Aggregate Function**

- These functions are used to do operations from the values of the column and a single value is returned.
- AVG() - Returns the average value COUNT() - Returns the
- number of rows FIRST() - Returns the first value
- LAST() - Returns the last value MAX() - Returns the largest value MIN() - Returns the smallest value SUM() - Returns the sum

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## SQL Statement Types

1. **AVG():**

Syntax: SELECT AVG(column_name) FROM table_name;
Example:
SELECT AVG(AGE) AS AvgAge FROM Students;

**2. COUNT()**

Syntax: SELECT COUNT(column_name) FROM table_name;
Example: SELECT COUNT(*) AS NumStudents FROM Stuents;

**3. FIRST()**

Syntax: SELECT FIRST(column_name) FROM table_name
Example:
SELECT FIRST(MARKS) AS MarksFirst FROM Students;

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## SQL Statement Types

**4. LAST()**
Syntax: SELECT LAST(column_name) FROM table_name;
Example: SELECT LAST(MARKS) AS MarksLast FROM Students;

**5. MAX()**
Syntax: SELECT MAX(column_name) FROM table_name;
Example :SELECT MAX(MARKS) AS MaxMarks FROM Students;

6. **MIN():** Similar to Max() we can use MIN() function.

**7. SUM() :**
Syntax: SELECT SUM(column_name) FROM table_name;
Example: SELECT SUM(MARKS) AS TotalMarks FROM Stuents**;**

TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

## SQL Statement Types

**PROCEDURE**

- A stored procedure is a prepared SQL code that you can save, so the code can be
- reused over and over again.
- So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.
- You can also pass parameters to a stored procedure, so that the stored procedure
- can act based on the parameter value(s) that is passed.
- To create procedure, use syntax:

```
CREATE PROCEDURE procedure_name AS sql_statement GO;
```

- To execute created procedure, use syntax:

```
EXEC procedure_name;
```

# SQL Statement Types

**Transaction Control**

- **The following commands are used to control transactions**

1. COMMIT – to save the changes.
2. ROLLBACK – to roll back the changes.
3. SAVEPOINT – creates points within the groups of transactions in which to ROLLBACK.

- **Commit:**
- The COMMIT command is the transactional command used to save changes invoked by a transaction to the database.
- The COMMIT command saves all the transactions to the database since the last
- COMMIT or ROLLBACK command.

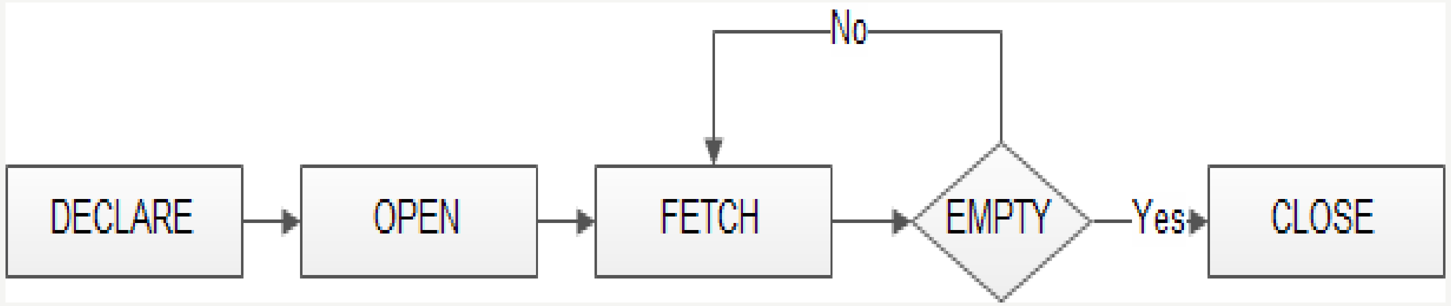
## SQL Statement Types

- **Rollback:**
- The ROLLBACK command is the transactional command used to undo transactions that have not already been saved to the database.
- This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.
- The syntax for a ROLLBACK command is as follows – ROLLBACK;

- **Savepoint:**
- A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.
- The syntax for a SAVEPOINT command is as shown below.
- SAVEPOINT SAVEPOINT_NAME;
- This command serves only in the creation of a SAVEPOINT among all the transactional statements. The ROLLBACK command is used to undo a group of transactions.

# SQL Statement Types

- **Cursor:**
- It is a temporary area for work in memory system while the execution of a statement is done.
- A Cursor in SQL is an arrangement of rows together with a pointer that recognizes a present row.
- It is a database object to recover information from a result set one row at once.
- It is helpful when we need to control the record of a table in a singleton technique, at the end of the day one row at any given moment. The arrangement of columns the cursor holds is known as the dynamic set.



# SQL Statement Types

- **Cursor:**

**Syntax:**

```
DECLARE variables;
records;
 create a cursor;
 BEGIN
OPEN cursor;
FETCH cursor;
process the records;
CLOSE cursor;
 END;
```

# SQL Statement Types

**Cursor Example:**
https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Cursor/example

**Stored Procedure with one parameter:**
https://github.com/TopsCode/Software-Engineering/tree/master/SQL/Cursor

**Stored Procedure with multiple parameter:**
https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Cursor/multipleParam

# SQL Statement Types

**Trigger**

- A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs
- For example, a trigger can be invoked when a row is inserted into a specified table.

**Syntax:**

create trigger [trigger_name] [before | after]
{insert | update | delete} on [table_name] [for each row] [trigger_body]

**Before Trigger:**

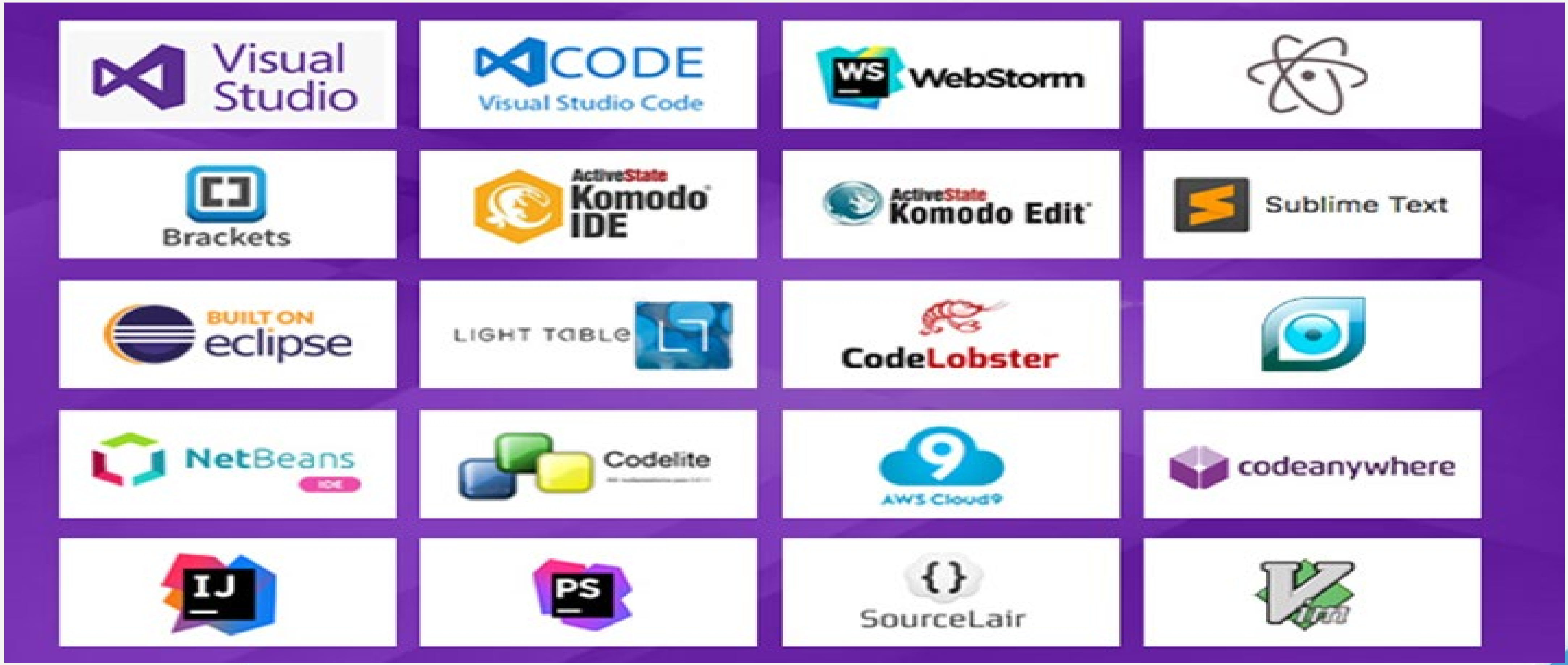
**https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Trigger/beforeTrigger**

**After Trigger:**

**https://github.com/TopsCode/Software-Engineering/blob/master/SQL/Trigger/afterTrigger**

## 11.Tools

## Tools can be used in Programing



TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

# THANK YOU