**DATABASE NOTES**

**----------------------------------------------------------------------------------------------------**

**~Sid’s**

**What is Database?**

A **database** is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, **databases** can be classified according to types of content: bibliographic, full-text, numeric, and images.

**What is Table?**

**Table** (**database**) ... In relational **databases** and flat file **databases**, a **table** is a set of data elements (values) using a model of vertical columns (identifiable by name) and horizontal rows, the cell being the unit where a row and column intersect. A **table** has a specified number of **columns, but can have any number of rows.**

**What is a Row?**

In the context of a relational **database**, a **row**—also called a record or tuple—represents a single, implicitly structured data item in a table. In simple terms, a**database** table can be thought of as consisting of **rows** and columns or fields.

**Inner Join :**

A relational database operation which selects rows from two tables such that the value in one column of the first table also appears in a certain column of the second table.

**Left outer join:**

A SQL **join** clause combines columns from one or more tables in a relational database. ... Join is a means for combining columns from one (self-table) or more tables by using values common to each.

**Right outer join:**

The **RIGHT JOIN** keyword returns all rows from the **right** table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.

**Group By:**

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

**Having:**

A **HAVING clause** in **SQL** specifies that an **SQL** SELECT statement should only return rows where aggregate values meet the specified conditions. It was added to the **SQL** language because the WHERE keyword could not be used with aggregate functions.

**Avg:**

Advertisements. **SQL AVG** function is used to find out the **average** of a field in various records. You can take **average** of various records set using GROUP BY**clause**.

**Select Top rows:**

The SELECT TOP clause is used to specify the number of records to return.

The SELECT TOP clause can be very useful on large tables with thousands of records. Returning a large number of records can impact on performance.

**Example for Max, sun, Avg**

SELECT SUM returns the sum of the data values.

SELECT AVG returns the average of the data values.

SELECT MAX returns the maximum value for a column.

*Syntax:*

*SELECT MAX(column-name) // Min*

*FROM table-name*

*SELECT SUM(column-name) // Sum*

*FROM table-name*

*SELECT AVG(column-name) // Avg*

*FROM table-name*

**Primary key:**

A Primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A primary key's main features are: It must contain a unique value for each row of data. It cannot contain null values.

**Foreign Key:**

In the context of relational databases, a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. In simpler words, the foreignkey is defined in a second table, but it refers to the primary **key** in the first table.

**Finding second highest salary from row table**

|  |  |
| --- | --- |
| Employee | |
| Employee ID | Salary |
| 3 | 200 |
| 4 | 800 |
| 7 | 450 |

select MAX(Salary) from Employee

SELECT MAX(Salary) FROM EMPLOYEE

WHERE Salary NOT IN (SELECT MAX (Salary) FROM Employee