Date: 12th Oct, 2023

**Union vs Union all: Create Two tables having same Name and datatype and same no of column and apply Union and union All and find what happen.**

**Union all:**

* Combine two tables into one table.
* Duplicate data will not be removed (i.e.). All data will be displayed.

Union

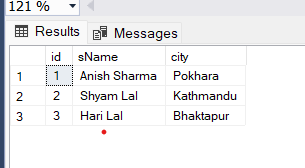
* Same as union all but here duplicated data will be removed or duplicated data will not be added into new table.

While using union or union all, we must select equal columns.

Eg: select column1, column2 from Table1 union/ union all select cloumn1, column2 from Table2.

Each column must have the same data types. Eg 🡪 int or varchar

**First Table: Students**

****

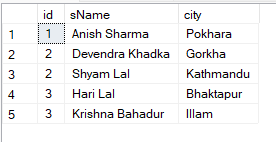
**Second Table: Teachers**

**A screenshot of a computer

Description automatically generated**

**Union all Result:**

select \* from Students union select \* from Teachers

****

**Union Result:**

select \* from Students union all select \* from Teachers

**A screenshot of a computer

Description automatically generated**

Normalization

* **Process of organizing data into database.**
* **Splitting table to reduce the redundancy data or duplication data.**
* **Insertion, deletion and updation anomaly are reduced.**

**1NF: First steps of normalization**

* **Should not contain multiple values at a column.**
* **Should contain same data type value.**
* **Each column should have a unique name.**

**2Nf: Second Steps of Normalization**

* **Should be in 1 NF.**
* **No partial dependency (in one table, column should not depend upon two or more)**

**3Nf:**

* **Should be in 2 NF.**
* **No transitive dependency (When one column depends on another which is non-prime attribute)**

**BCNF: Boyce-Codd Normal Form**

* **Advance version**
* **Should not depend on super key.**

**4Nf:**

* **Should be in BCNF.**
* **No Multi-value dependency**

**5Nf:**

* **Should be on 4Nf.**
* **No Join Dependency**

**De-Normalization**

* **Process to add redundant data into two or more tables.**
* **Retrieving data is faster.**
* **Less use of Joins.**
* **But Insertions and deletion can be complex and more cost.**
* **Doesn’t mean Reversing the normalization.**

**When Normalization and De-Normalization**

* Normalization, at transactions like Insertion, Updation, Deletion.
* De-normalization, at the data retrieving because it reduces joins, means data retrieving will be fast.

**Concept of Keys**

* Keys are the use to represent the columns or tuples.
* It is also used to establish the relationship between the tables or columns.

**Primary Key:**

* Used to identify the column uniquely.
* One table has only one primary key.

**Super Key:**

* **Also represent the columns or table uniquely.**
* **Primary keys are selected from the super key.**

**Foreign Key:**

* **Used to connect the different entity or tables.**

**Composite Key:**

* **Set of multiple attributes.**
* **Also refers to the unique tuples.**

**Alternate Key:**

* **Those keys which are selected for primary key but not a primary key.**

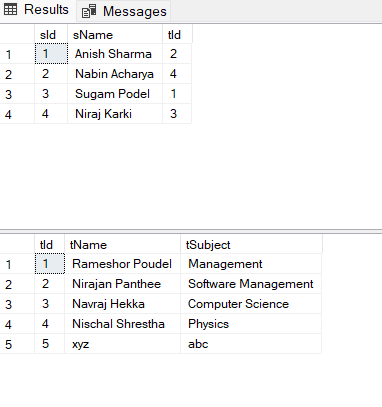
**Unique Key:**

* **Same as primary key but it can have nullable value.**

**Sql JOIN**

* **Join clause is used to combine two or more columns of a table.**

**For e.g.: We have two tables.**

****

**Inner Join:**

* **Only show the matching data from two tables**
* select sName, tName from students inner join teachers on students.tId = teachers.tId

**Result:**

**A screenshot of a computer

Description automatically generated**

**Left Join:**

* **Show all the data from left, and show matching data from right.**
* select students.sName, teachers.tName from students left join teachers on students.tId = teachers.tId

Result:

**A screenshot of a computer

Description automatically generated**

**Right Join**

* **Show all the data from right, also show matching data from left.**
* select students.sName, teachers.tName from students right join teachers on students.tId = teachers.tId

**Result:**

**A screenshot of a computer

Description automatically generated**

**Full Outer Join:**

* **Show all the data from left as well as from right.**
* select students.sName, teachers.tName from students full outer join teachers on students.tId = teachers.tId order by teachers.tName

**Result:**

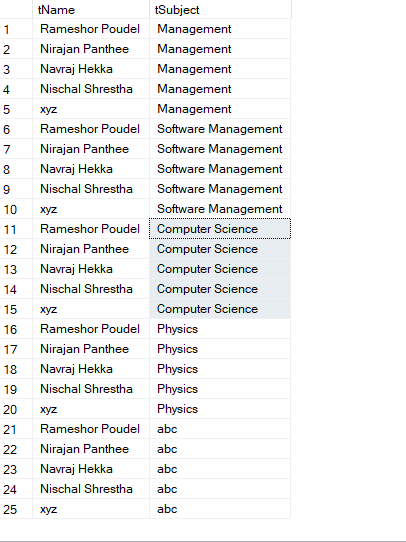
**A screenshot of a computer

Description automatically generated**

**Self-Join:**

* **Join to itself.**
* **Not used in real life**
* select t1.tName, t2.tSubject from teachers t1, teachers t2

Result:

****

Date: 15th Oct, 2023

Q. Suppose you have a table having column EmployeeId,EmployeeName,ManagerId then to find the manager for each employee which join you are going to apply and write the query also.

Ans:

I will use left join.

* Query: select Employee.EmployeeName, Manager.ManagerName from Employee left join Manager on Employee.tId = Manager.tId

**Where vs Having**

* **Where clause can be use without ‘Group By’ in other hand, having cannot be used without ‘Group By’,**
* **Where can be used with DELETE, UPDATE, and SELECT statements in other hand, Having is only used with SELECT statement.**
* **Where doesn’t contains any aggregate function in other hand, Having contains aggregate function.**

**Example:**

Where:

select EmployeeName, Salary from Employees where Salary >= 5000;

Having:

select sum(Salary) as 'Total Salary' from Employees group by Salary having salary > 50000

**Q. Can Primary key be null?**

**🡪 Primary key used to uniquely identify the data of columns. If we put the primary key to NULL, then it won’t identify the column or table.**

**Therefore, Primary key cannot be null.**

**Q. Can Foreign key be duplicated?**

**🡪 Yes, because foreign key doesn’t need to store unique value and it can contain null value.**

**Refertential Integrity Constraints**

* **It is a constraint on database that each foreign key from one table, points the unique primary of another table.**
* **It is also known as Primary Key Constraints.**
* **It is used to create a relationship between two tables.**
* **First table is Master table which much have reference column which have primary key or unique key.**
* **Second table is Child key where we define the foreign key column. It must have value which have referred key in master table.**

**Q. Create a table having column ( Id(int,Primarykey),Name(varchar(50),IsActive(bit),Amount(double)) and apply Default Constraints in IsActive Column and set a Default Value 1,andCreate a Check Constraints in Amount field to restrict user from entering value Less Than 0.**

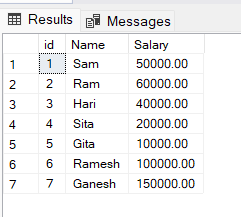
**🡪** create table Employee (EId int primary key not null identity, EmpName varchar(50),

IsActive bit default 1,

Amount decimal(10,2), check (Amount > 0))

**Q. Create a table having column ( Id(int,Primarykey),Name(varchar),Salary(double)) Select name of the employee and position by applying condition if the salary>1Lakh then position is 'Senior',50K< Salary<1Lakh Then position is 'MidLevel' else 'Junior'**

* **Table:**

****

* **Query:**

select Name, case

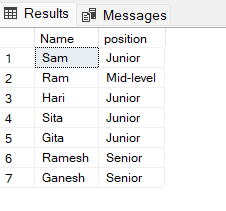
when Salary >= 100000 then 'Senior'

when salary > 50000 and Salary < 100000 then 'Mid-level'

else 'Junior'

end

as 'position' from Employee

****

**Q.** Create a custom Function by yourself and execute it.

🡪 create function ShowEmpById (@Id int)

returns varchar(100)

as

begin

declare @Name varchar(100)

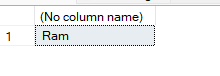
select @Name = EmpName from Employee

where EId = @Id

return @Name

end

* After Executing : select dbo.[ShowEmpById] (1)



**Date: Oct 16th**

**Stored Procedure:**

* **It’s a prepared code that has been saved by us.**
* **It helps to reuse the code over and over again.**
* **We have to call the procedure to execute it.**

**Query:**

create procedure SelectAll

as

select \* from TbEmployee

go

To execute query:

exec SelectAll

**Different between Stored Procedure and the function**

* **Stored procedure is executed only once, and it can be used by calling that procedure\_name while in function when we call the function it first execute and return the value or call.**
* **Function must return the value but it is optional in procedure.**