

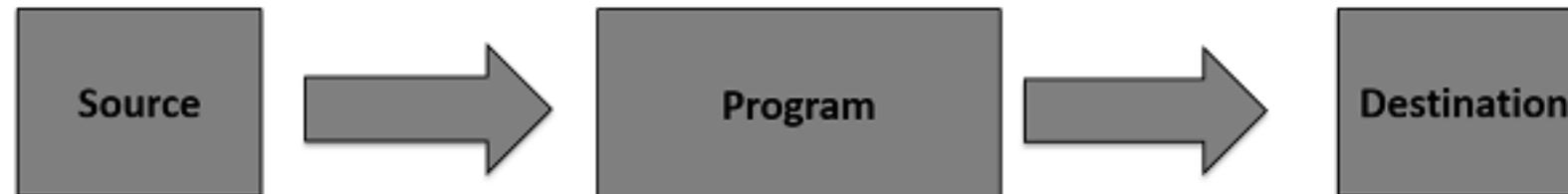
BridgeLabz

Employability Delivered

MSSQL DB

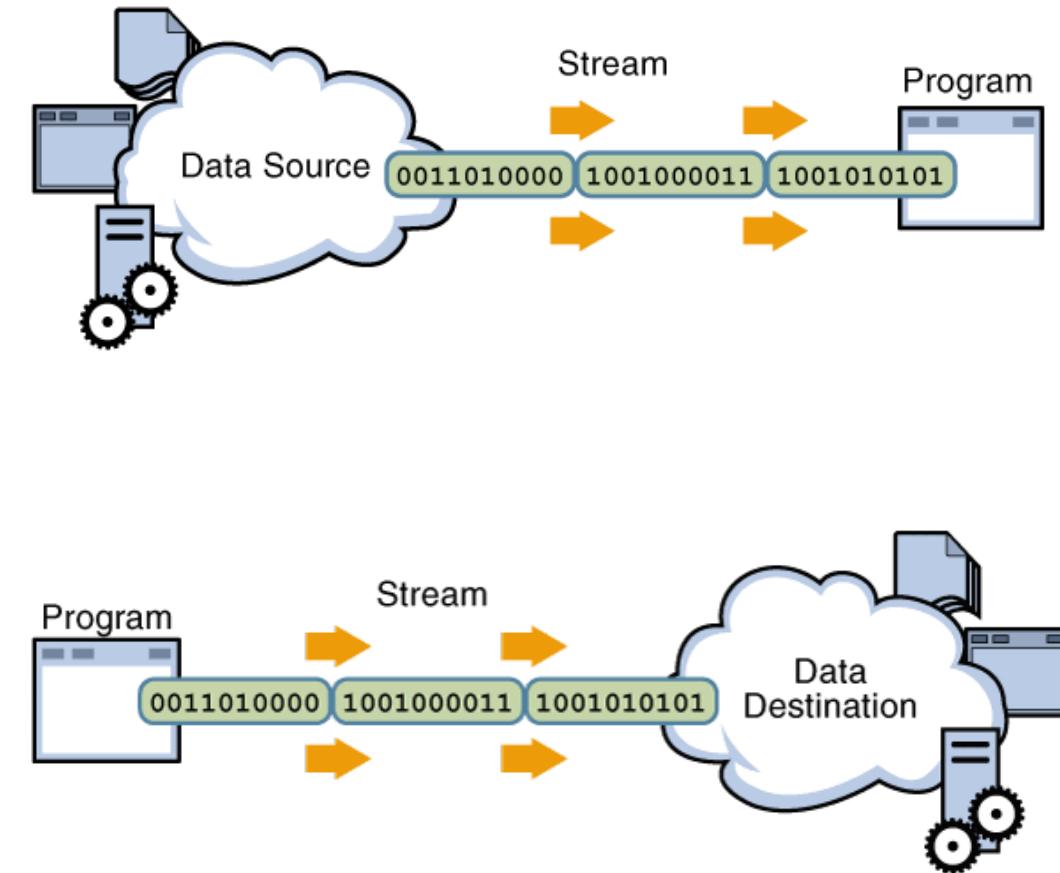
IO – Input and Output Streams

- Typically IO streams represent an input source to read data and an output destination to write data.
- A stream can be defined as a sequence of data. There are two kinds of Streams –
 - **Input Stream** – The Input Stream is used to read data from a source.
 - **Output Stream** – The Output Stream is used for writing data to a destination.
- The Source and Destination can be
 - Console
 - File (`system.io`)
 - Database (`Ado.Net`)
 - Cloud Server (`REST`)
- The Data can be a **byte stream**, **Ado.Net format**, **CSV Format**, and **JSON Format** etc.



IO – Input and Output Streams

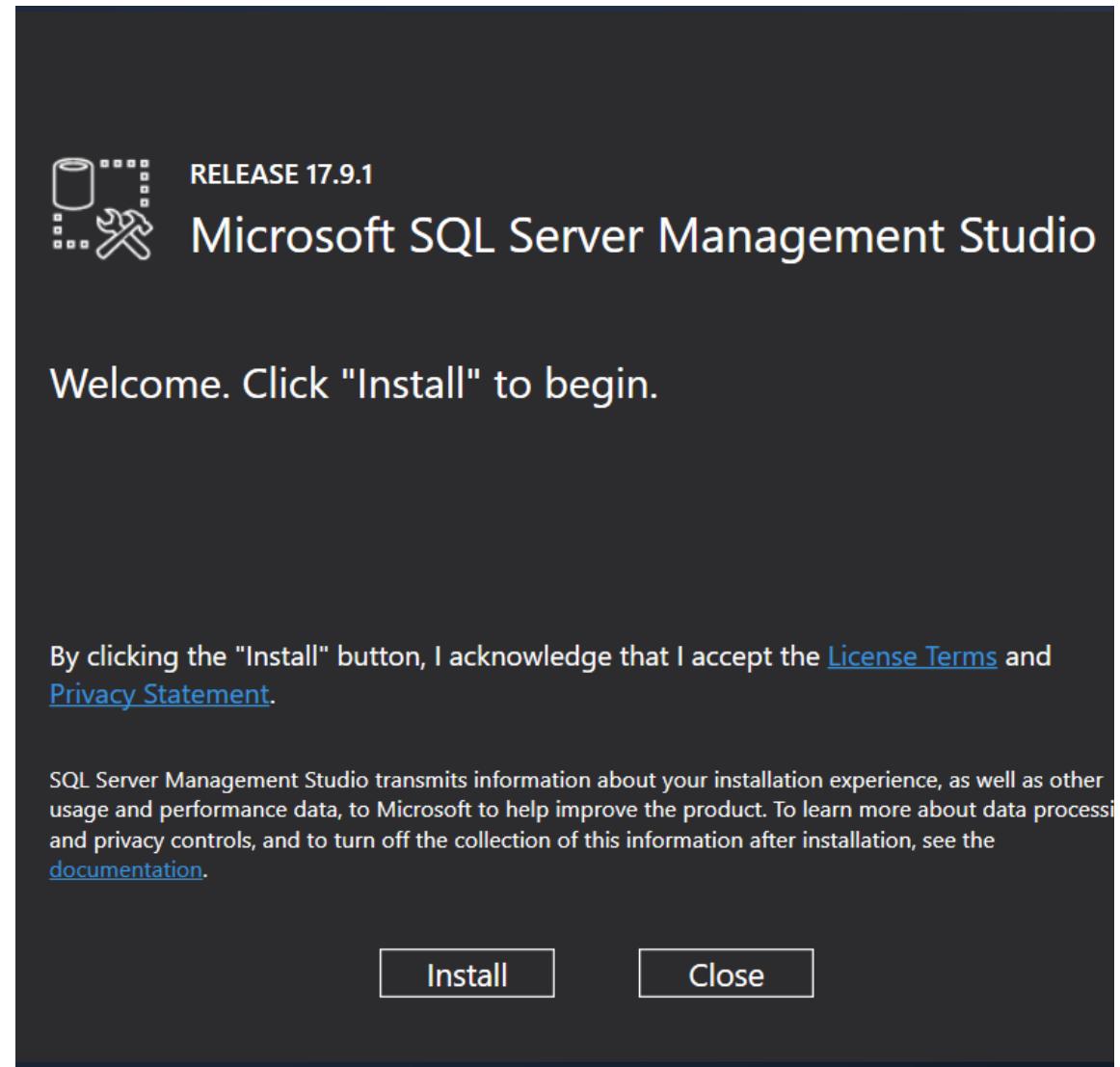
- **Input Stream** – Input streams are used to read the data from various input devices like keyboard, file, network, etc.
- **Output Stream** – Output streams are used to write the data to various output devices like monitor, file, network, etc.



Section 1: MSSQL DB

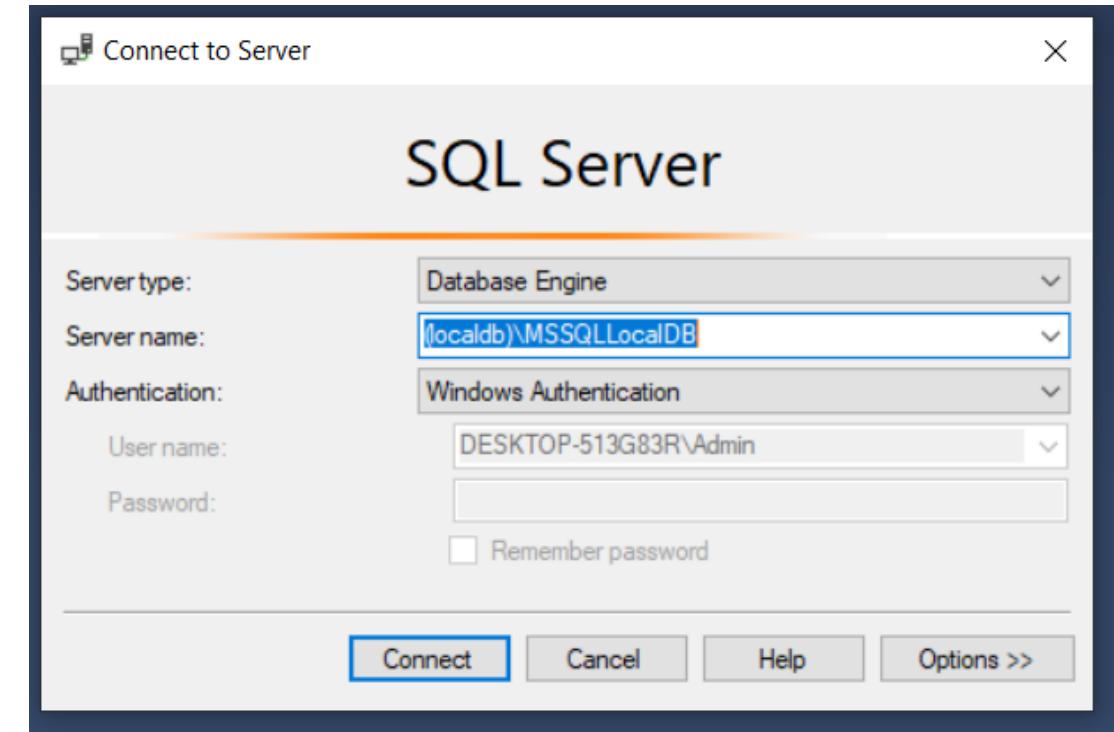
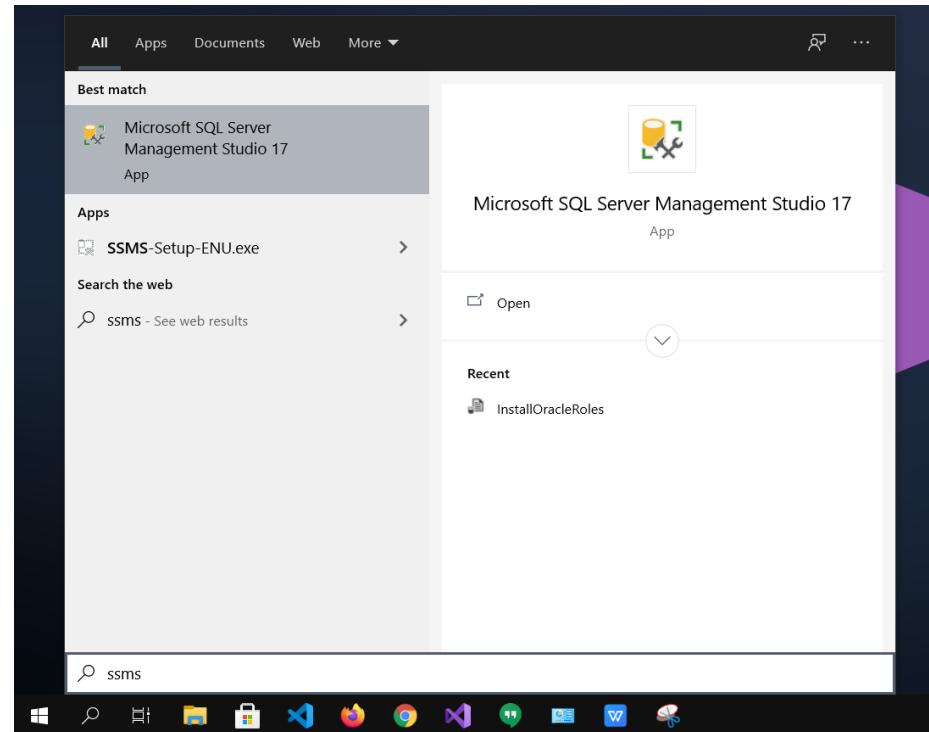
MSSQL & SSMS 2017 Download

- Install MS SQL Server from
<https://go.microsoft.com/fwlink/?LinkId=2043154>
- Direct Download SSMS 2017 Developer Version from the above link - Double click **SSMS-Setup-ENU.exe**
- *If you receive a User Account Control notice, choose Yes.*



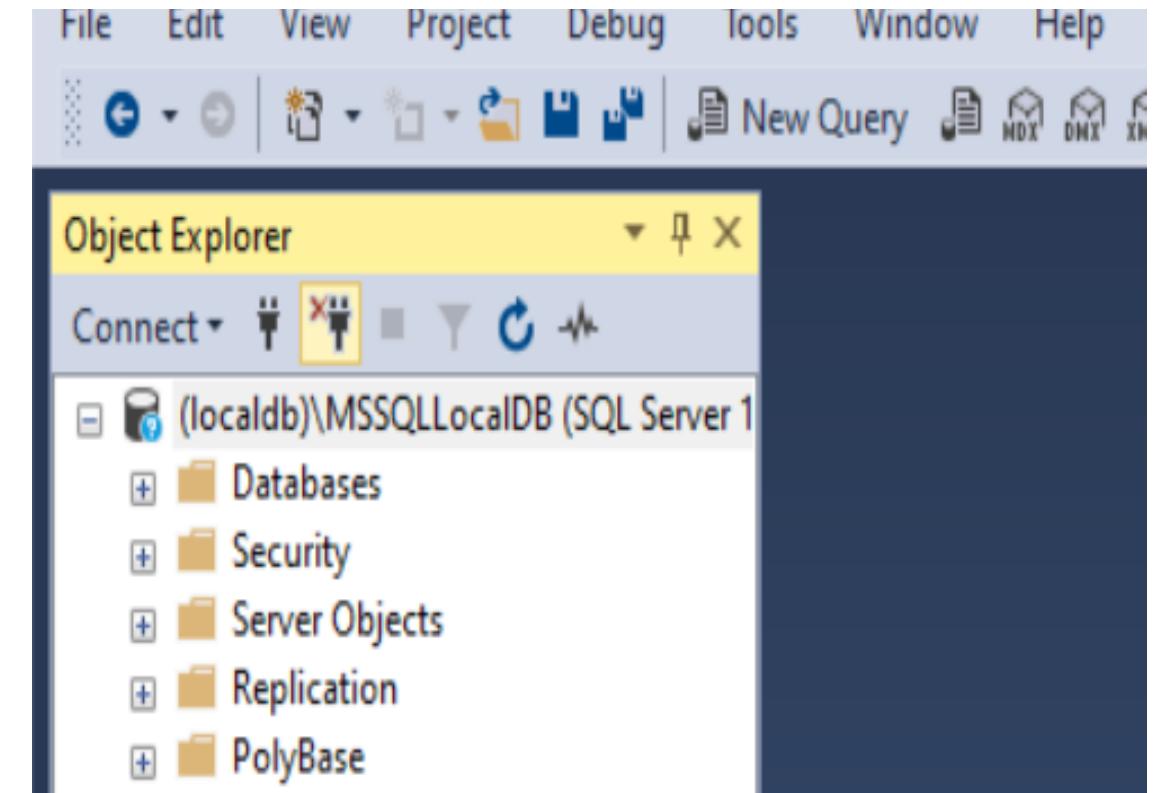
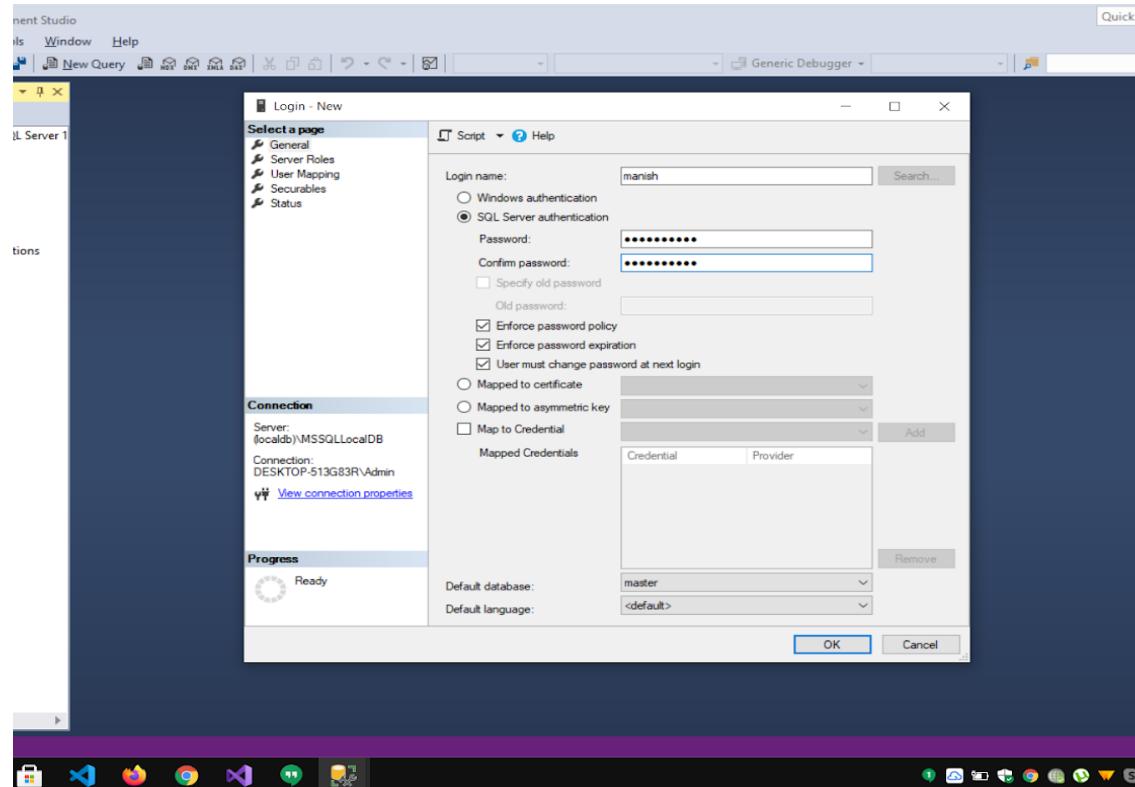
MS SQL Settings

- Once the SSMS got installed Click on Close.
- Press Windows key and search for SSMS and click to open as shown below.
- Once SSMS is opened we need to give a **Server name → Connect** to access SSMS and work on the database.



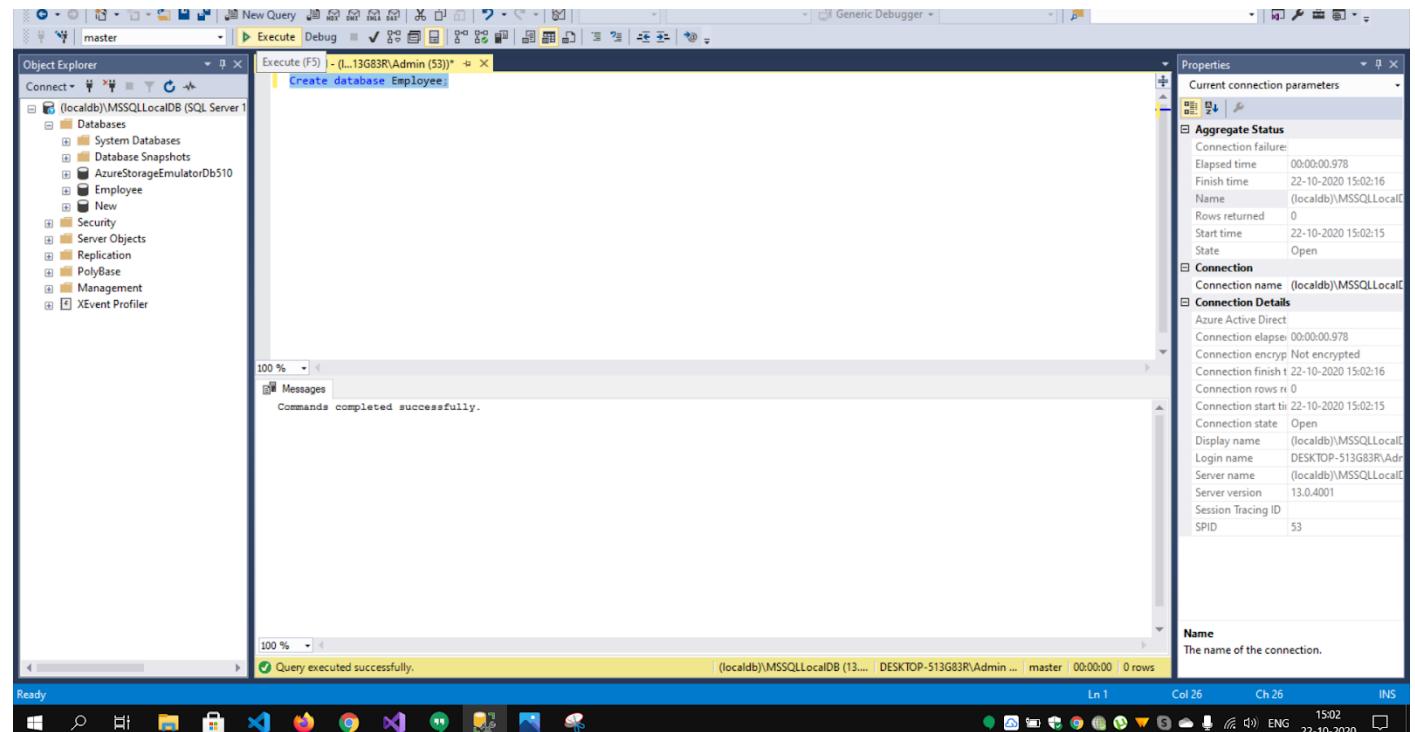
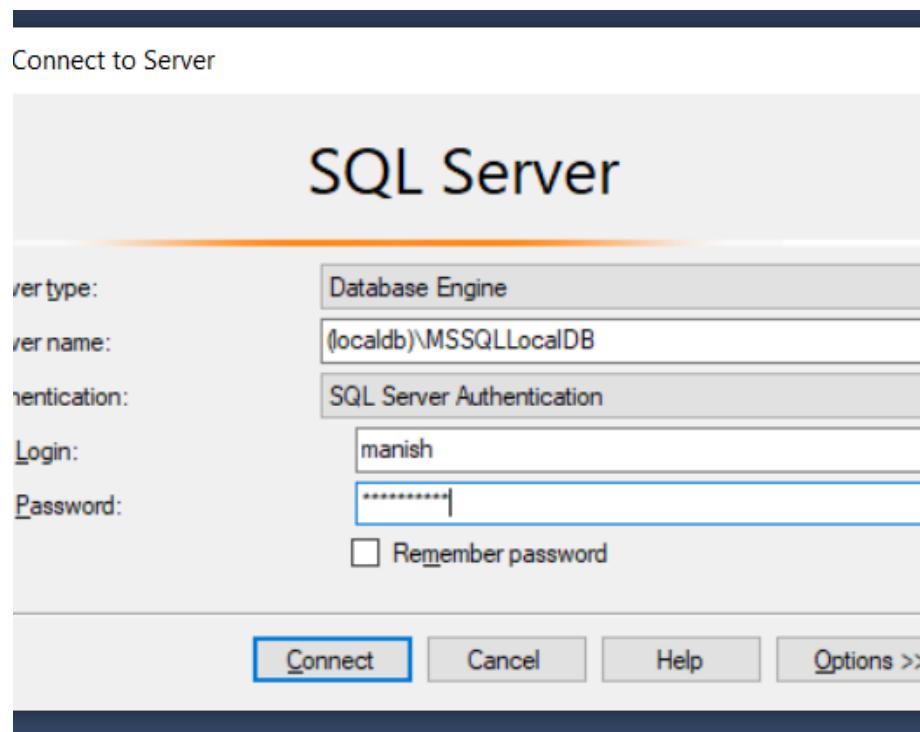
MS SQL Settings

- As click on login will get a popup page to add the Security such as **Login and Password → Ok** as shown below.
- Verify whether SQL authentication is working or not by following the below steps.
- Disconnect from the database.



MS SQL Settings

- Change the Authentication from **Windows Authentication** to **SQL Server Authentication**.
- Enter **Login** and **Password** we had created in the above step as shown below.(**Note** :- Server name is optional).
- Click on **New Query (or) CTRL + N** to open the Query page, where we can write our SQL queries and execute them by clicking **Execute (or) F5**



Key Definitions

- **Database** – Database is a organized collection of data. For e.g. Employee Payroll Service Database to manage payroll of all employees
- **Database Management System (DBMS)** – A DBMS is a software application which is used for managing different databases which essentially involves tasks like Data Security, Data Backup, Data Export/Import, Manage Huge Data, Give access to programs, etc.
- **Types of Databases –**
 - **Relational Database** – In relational database, data is organized in form of tables. A table contains rows and columns of data. Table has a unique key to identify each row of the table.
 - **Non-relational databases** – Data is not organized in form of tables. Data is stored in form of key & value pairs like in JSON Format. Also call **NoSQL DB**. Most popular is **Mongo DB**. You will get some exposure using **JSONServer**
- **SQL** is a Structured Query Language and is used to interact with relational databases. We often refer relational database as **SQL DB**.
- **Query** – A Query is a set of instruction given to the database management system, which tells RDBMS what information you would like to get from the database.

What is SQL?

- SQL stands for Structured Query Language, which is a standardized language for interacting with **RDBMS** (Relational Database Management System).
- Some of the popular relational database example are: MySQL, Oracle, PostgreSQL, Microsoft SQL Server, IBM DB2 etc.
- We use SQL Client to perform Database Operation which includes creating databases and tables inside database.
- SQL is also used to perform **C.R.U.D** operations on relational databases.
 - Create – We use **Insert** Queries to create rows in DB
 - Retrieve – We use **Select** Queries to retrieve the rows in the DB
 - Update – We use **Update** Queries to update the rows in the DB
 - Delete – We use **Delete** Queries to delete the rows in the DB
- NOTE: SQL Statements are Case Insensitive and the statement ends with “;” semicolon to indicate the DB that the Query Statement is complete. For e.g. `SELECT * FROM employee_payroll;`
- SQL can also perform administrative tasks on database such as database security, backup, user management etc.



UC 1

Ability to create a payroll service database

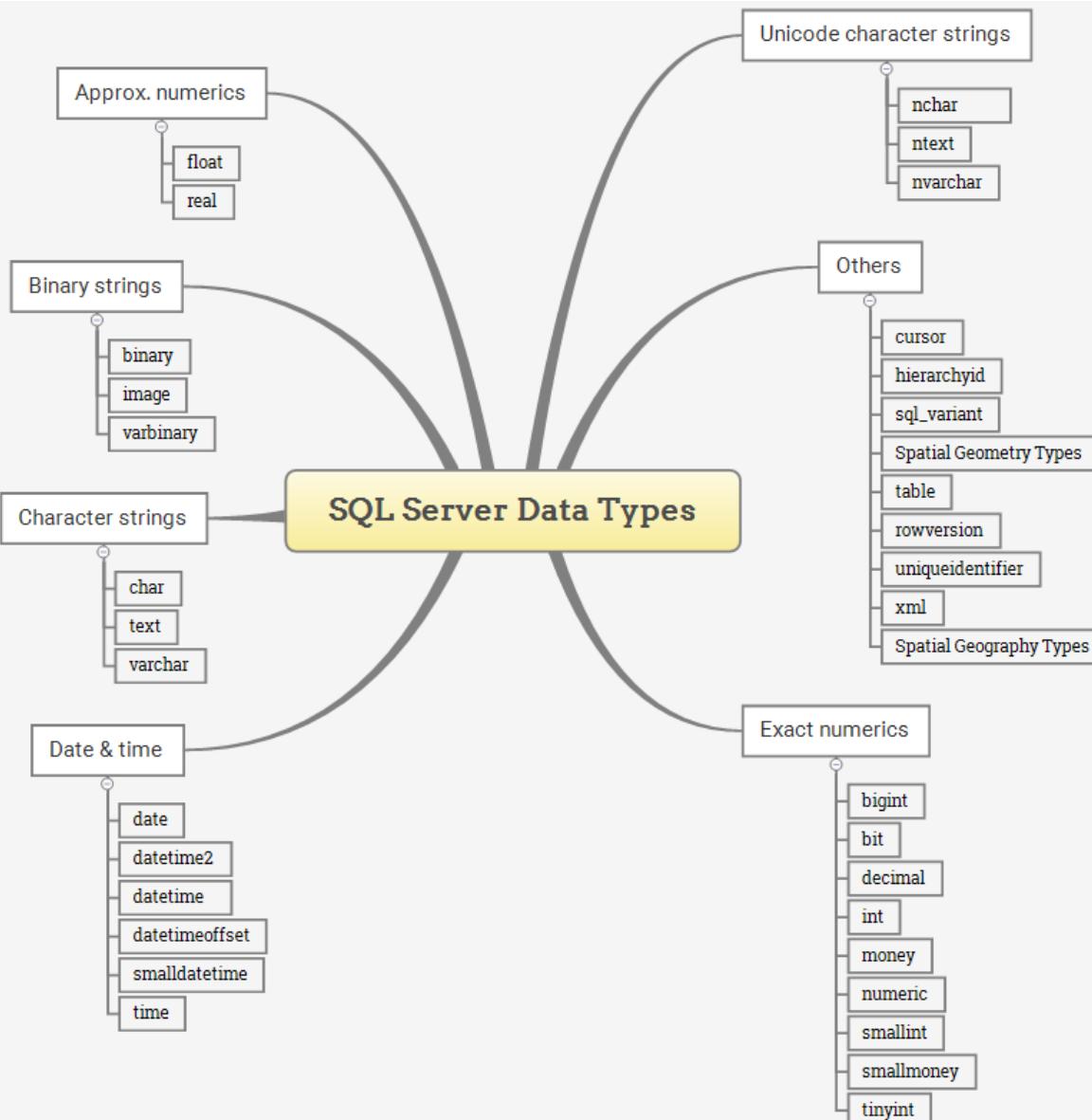
- Using MSSQL Client use `create database` query to create a `payroll_service` database
- Also you can see the DB created by using `show database` query
- And go to the database created by using `use payroll_service` query



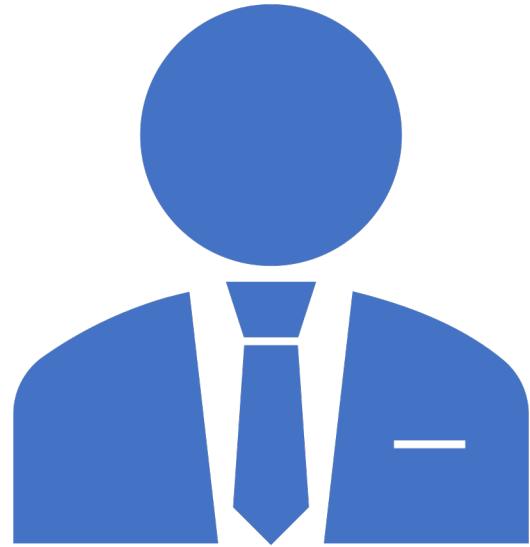
UC 2

Ability to create a employee payroll table in the payroll service database to manage employee payrolls

- Use payroll_service database in MSSQL Client
- Use `Create Table employee_payroll` Query to create employee payroll table with columns id, name, salary and start date as column. Note Id is set to auto increment.
- Understand the SQL data types to be used for the table
- Note: SQL Queries as case insensitive



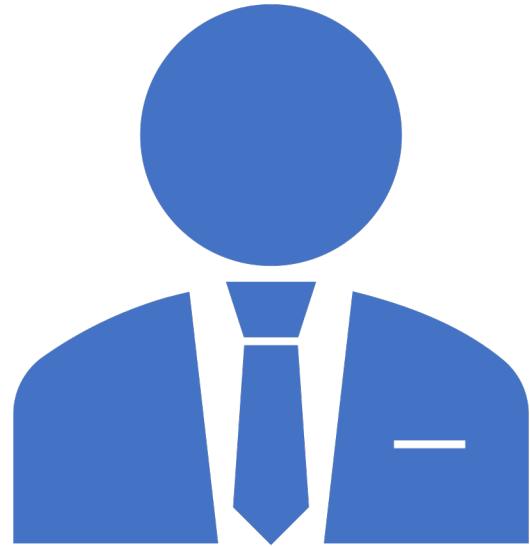
MSSQL Data Types



UC 3

Ability to create employee payroll data in the payroll service database as part of **CURD Operation**

- Use payroll_service database in MSSQL Client
- Use `INSERT INTO employee_payroll` Query to create employees payroll data into the employee_payroll table



UC 4

Ability to retrieve all the employee payroll data that is added to payroll service database

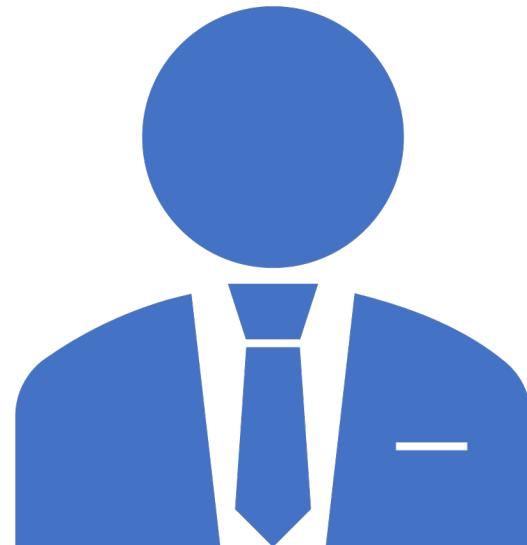
- Use payroll_service database in MSSQL Client
- Use `SELECT * FROM employee_payroll` to retrieve all the data from the employee_payroll table



UC 5

Ability to retrieve salary data for a particular employee as well as all employees who have joined in a particular date range from the payroll service database

- Use `SELECT salary FROM employee_payroll WHERE name = 'Bill'` Query to View Bill's salary
- Use `Select` query with `Where` condition View employees between start dates
- Query: `WHERE start BETWEEN CAST('2018-01-01' AS DATE) AND DATE(NOW())`
- Note: Where Condition Clause is used to retrieve the row needed from the table
- Note: Use of Database Functions like `CAST()` and `NOW()` in the Query



UC 6

Ability to add Gender to Employee Payroll Table and Update the Rows to reflect the correct Employee Gender

- Use payroll_service database in MSSQL Client
- Use Alter Table Command to add Field gender after the name field
- Use Update Query to set the gender using where condition with the employee name
- E.g. `UPDATE employee_payroll set gender = 'M' where name = 'Bill' or name = 'Charlie';`



UC 7

Ability to find sum, average, min, max and number of male and female employees

- Use payroll_service database in MSSQL Client
- Use Database Function **SUM, AVG, MIN, MAX, COUNT** to do analysis by Male or Female.
- Note: You will need to use GROUP BY GENDER grouping to get the result
- E.g. **SELECT SUM(salary) FROM employee_payroll WHERE gender = 'F' GROUP BY gender;**

MSSQL Commands

```
1 CREATE TABLE employee_payroll
2 (
3     EmployeeID int identity(1, 1) primary key,
4     EmployeeName varchar(255),
5     PhoneNumber varchar(255),
6     Address varchar(255),
7     Department varchar(255),
8     Gender char(1),
9     BasicPay float,
10    Deductions float,
11    TaxablePay float,
12    Tax float,
13    NetPay float,
14    StartDate Date,
15    City varchar(255),
16    Country varchar(255)
17 );
18
19 insert into employee_payroll
20 (EmployeeName,PhoneNumber,Address,Department,Gender,BasicPay,Deductions,TaxablePay,T
21 values
22 ('Gunjan', '7878787878', 'Mumbai', 'ENGG', 'M', 30000, 2000, 1000, 200, 18000, getd
23
24 select * from employee_payroll
25
26 select * from employee_payroll where EmployeeName='Terisa'
27
28 select schema_name(t.schema_id) as schema_name,
29         t.name as table_name,
30         t.create_date,
31         t.modify_date
32     from sys.tables t
33     order by schema_name,
34             table_name;
```

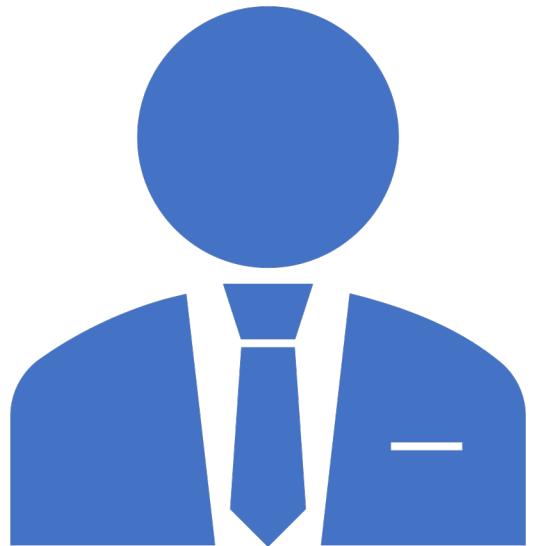
Section 2: ER Diagram



UC 8

Ability to extend
employee_payroll data to
store employee information
like employee phone, address
and department

- Ensure employee department is non nullable fields.
- Add Default Value for address field.



UC 9

Ability to extend
employee_payroll table
to have Basic Pay,
Deductions, Taxable Pay,
Income Tax, Net Pay

Employee Payroll Table

id	name	phone_number	address	department	gender	basic_pay	deductions	taxable_pay	tax	net_pay	start
2	Terisa	NULL	NULL		F	3000000	0	0	0	0	2019-11-13
3	Charlie	NULL	NULL		M	3000000	0	0	0	0	2020-05-21
4	Bill	NULL	NULL		M	1000000	0	0	0	0	2018-01-03
5	Mark	NULL	NULL	TBD	NULL	1000000	0	0	0	0	2018-01-03



UC 10

Ability to make Terissa as part of Sales and Marketing Department

- Note: The Complete employee payroll roll need to be Inserted
- If a Salary is now going to be updated multiple rows has to be updated creating unnecessary redundancy
- Further There is 2 Employee ID so according to Database there is two different Terissa

Employee Payroll Table

```
INSERT INTO employee_payroll
(name, department, gender, basic_pay, deductions, taxable_pay, tax, net_pay, start) VALUES
('Terisa', 'Marketting', 'F', 3000000.00, 1000000.00, 2000000.00, 500000.00, 1500000.00, '2018-01-03' );
```

25

26 select * from employee_payroll where EmployeeName='Terisa'

Results Messages

	EmployeeID	EmployeeName	PhoneNumber	Address	Department	Gender	BasicPay	Deductions	TaxablePay	Tax	NetPay	StartDate	City	Country
1	1	Terisa	78787878787	Mumbai	HR	F	20000	2000	1000	200	18000	2020-10-22	Mumbai	IN

Normalization and Denormalization of DB

- Normalization is the technique of dividing the data into multiple tables to reduce data redundancy and inconsistency and to achieve data integrity.
- Normalization is a strategy used on a previously denormalized database to increase integrity.
- Here are the most commonly used normal forms:
 - First normal form(1NF)
 - Second normal form(2NF)
 - Third normal form(3NF)
 - Boyce & Codd normal form (BCNF)
- Denormalization is the technique of combining the data into a single table to make data retrieval faster.
- Denormalization is a strategy used on a previously-normalized database to increase read performance of a database, at the expense of losing some write performance, by adding redundant copies of data or by grouping data.

Normalization Forms

1NF - Each attribute of a table must have atomic (single) values

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
			8812121212
102	Jon	Kanpur	9900012222
			7778881212
103	Ron	Chennai	9990000123
104	Lester	Bangalore	8123450987

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
102	Jon	Kanpur	9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123
104	Lester	Bangalore	8123450987

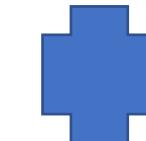
2NF – Non Prime attribute should be dependent on all Prime Attributes

teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40



teacher_details table

teacher_id	teacher_age
111	38
222	38
333	40



teacher_details table

teacher_id	subject
111	Maths
111	Physics
222	Biology
333	Physics
333	Chemistry

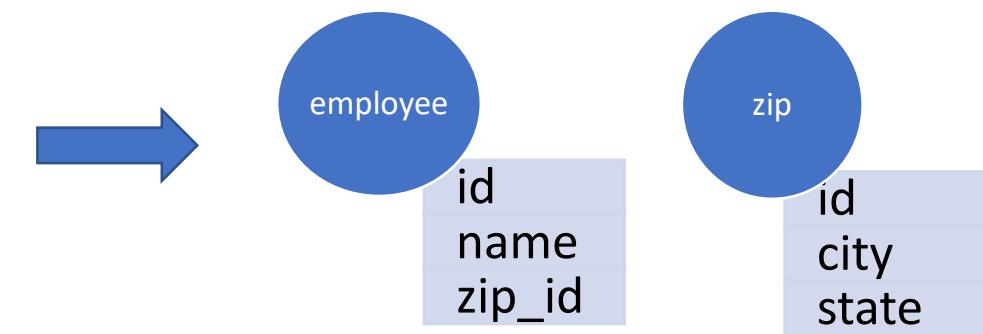
Candidate Keys: {teacher_id, subject}

Non prime attribute: teacher_age

Normalization Forms

3NF – Transitive functional dependency of non-prime attribute on any super key should be removed.

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

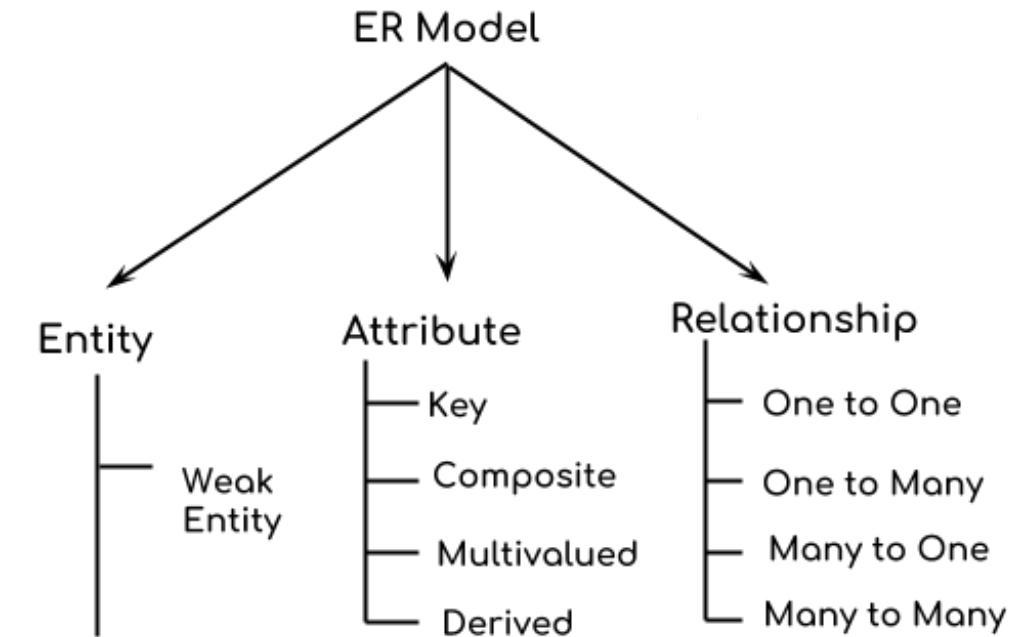


Boyce Codd normal form (BCNF) –

BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X should be the super key of the table.

Entity Relationship Diagram – ER Diagram

- An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**.
- An ER model is more intuitive approach than following rules in Normalized forms like 1NF, 2NF, etc. and is a design or blueprint of a database that can later be implemented as a database. Very similar to Class Diagram.
- The best way to design the DB is to keep the knowledge of Normalized Forms and use the Intuitive Approach and Domain Knowledge to arrive at the ER Diagram.
- The main components of E-R Diagram are:
 - Entity
 - Attribute
 - Relationship



Components of ER Diagram



Employee Payroll Table



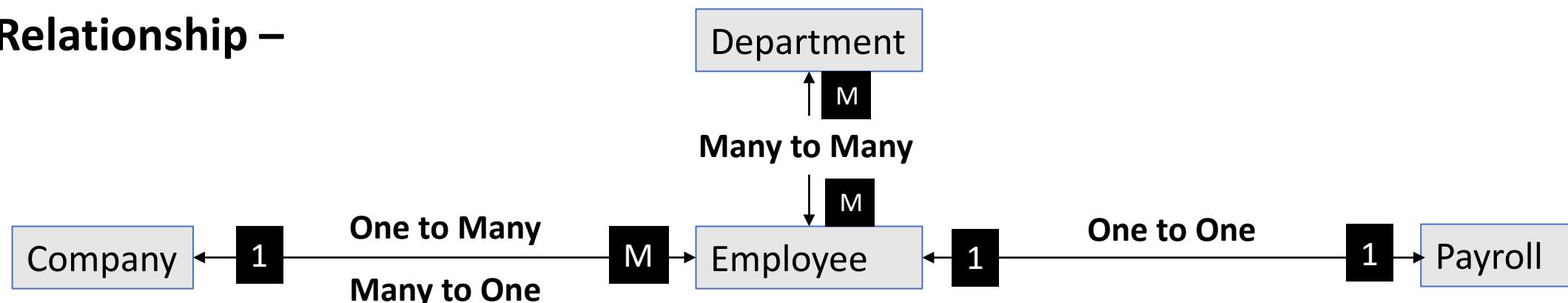
UC 10

Draw the ER Diagram for Payroll Service DB

- Identifies the Entities – Entities can be Identified using Normalization Technique
- Check each attribute and see if they are Composite or Multi-Valued

Entity Relationship Diagram – ER Diagram

- **Entity** – An Entity is similar to a Class or a Table in DB and is identified by its attributes and relationship with other entities. An Entity can be a **Weak Entity** if it dependent on other Entity.
- **Attribute** – An Attribute is the property of an Entity and are following 4 Types:
 - **Key attribute** – Uniquely Identified the Entity e.g. **employee id**
 - **Composite attribute** – is a combination of other attributes e.g. **employee address**
 - **Multivalued attribute** – hold multiple values like the **phone number**
 - **Derived attribute** – value is dynamic and derived from another attribute e.g. **net pay**
- **Relationship** –

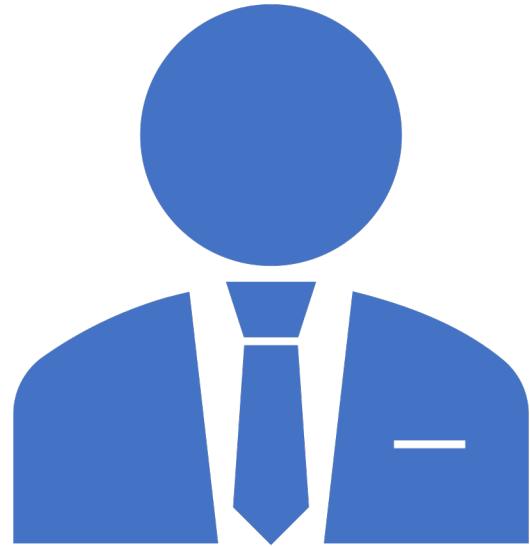




UC 11

Implement the ER Diagram into Payroll Service DB

- Create the corresponding tables as identified in the ER Diagram along with attributes
- For Many to Many relationship, create new Table called Employee Department having Employee Id and Department ID and redo the UC 7

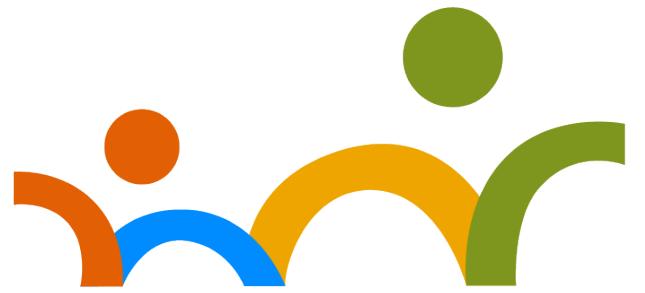


UC 12

Ensure all retrieve queries done especially in UC 4, UC 5 and UC 7 are working with new table structure

References

- [Introduction to SQL](#)
- [Basic SQL Queries](#)
- [DB Normalization](#)
- [ER Diagram Concept Introduced for starters](#)



BridgeLabz

Employability Delivered

Thank
You