Doubts

1. Should we check dob < date of joining And other obvious things
2. Check for AUTO-INCREMENT

Size keys Name(50), ID(20), Tag(20), Language(20), Email(100)

RELATIONAL SCHEMA

USER

ID	VarChar(20) NotNull Unique
TrustRating	Int Default = 100 NotNull
Email	VarChar(100) Unique NotNull
UserName	VarChar(50) Unique NotNull
First_Name	VarChar(50) NotNull
Last_Name	VarChar(50) NotNull
Location_City	VarChar(50) NotNull
Location_Country	VarChar(50) NotNull
DateOfJoining	Date NotNull
DateOfBirth	Date NotNull

Organization	VarChar(50)

Constraints:

- **Primary Key →** ID
- **Check →** (Date of joining > Date of birth)
- **Derived →** (Solved Hard, Solved Medium, Solved Easy, User Experience Strength, User Experience Weakness, Followers)

USER_LANGUAGES

ID	VarChar(20) NotNull
Language	VarChar(20) NotNull

Constraints:

- **Primary Key→** (ID, Language)
- **Foreign Key →** ID references USER(ID)
- **Check →** Language in {C++, C, Python, Java, JS, Kotlin, C#}

PREMIUM USER

ID	VarChar(20) NotNull Unique
ProfileVisits	Int Default = 0
Subscription_TimeStart	Date Not Null
Subscription_TimeEnd	Date Not Null

Constraints:

- **Primary Key →** ID
- **Foreign Key →** ID references USER(ID)

- **Check ⟶** (Subscription Time Start < Subscription Time End)
- **Check ⟶** (Date of joining > Date of birth)
- **Derived ⟶** (AmountPaid)

REPOSITORY

Name	Varchar(60) Not Null
Date	Date Not Null
ID	VarChar(20) NotNull

Constraints:

- **Primary Key ⟶** (ID, Name)
- **Foreign Key ⟶** ID references USER(ID)

REPO_TEMPLATES

ID	VarChar(20) NotNull
Name	Varchar(60) Not Null
Template_Name	VarChar(50) NotNull
Template_Language	VarChar(20)
Template_Content	Long Text

Constraints:

- **Primary Key ⟶** (ID, Name, Template_Name)
- **Foreign Key ⟶** (ID, Name) references REPOSITORY(ID, Name)
- **Check ⟶** Template_Language in {C++, C, Python, Java, JS, Kotlin, C#}

REPO_TAGS

ID	VarChar(20)

	NotNull
Tag	VarChar(20) NotNull
Name	Varchar(60) Not Null

Constraints:

- **Primary Key** ➞ (ID, Tag, Name)
- **Foreign Key** ➞ (ID, Name) references REPOSITORY(ID, Name)
- **Check** ➞ Tag in {graphs, DP, binary search, greedy, implementation, data structures, brute force, math, strings, number theory}

PROBLEMS

Name	Varchar(50) NotNull
Problem_ID	Int NotNull Unique Auto Increment
Rating_Difficulty	VarChar(20) NotNull
ID	VarChar(20) NotNull

Constraints:

- **Primary Key** ➞ Problem_ID
- **Foreign Key** ➞ ID references PROGRAMMING_ORGANIZATION(ID)
- **Derived** ➞ Solves

PROBLEMS_TAGS

Problem_ID	Int NotNull
Tag	VarChar(20) NotNull

Constraints:

- **Primary Key ➞** (Problem_ID**,** Tag)
- **Foreign Key ➞** Problem_ID references PROBLEMS(Problem_ID)
- **Check ➞** Tag in {graphs, DP, binary search, greedy, implementation, data structures, brute force, math, strings, number theory}

BLOGS

Name	Varchar(50) NotNull
Date	Datetime Not Null
Likes	Int Default = 0
Content	Long Text NotNull
ID	VarChar(20) NotNull

Constraints:

- **Primary Key ➞** (ID, Name, Date)
- **Foreign Key ➞** ID referenced from USER(ID)

BLOGS_TAGS

ID	VarChar(20) NotNull
Tag	VarChar(20) NotNull
Name	Varchar(60) Not Null
Date	Datetime Not Null

Constraints:

- **Primary Key ➞** (ID, Tag, Name, Date)
- **Foreign Key ➞** (ID, Name, Date) references BLOGS(ID, Name, Date)
- **Check ➞** Tag in {graphs, DP, binary search, greedy, implementation, data structures, brute force, math, strings, number theory}

CONTEST

Name	Varchar(50) NotNull
Date	Datetime Not Null
Likes	Int Default = 0
ID	VarChar(20) NotNull
DateOfContest	Datetime Not Null
Content	Long Text

Constraints:

- **Primary Key ➞** (ID, Name, Date)
- **Foreign Key ➞** ID references PROGRAMMING_ORGANIZATION(ID)
- **Check ➞** Tags in (graphs, DP, binary search, greedy, implementation, data structures, brute force, math, strings, number theory)
- **Check ➞** (Date of Contest > Date)

CONTEST_TAGS

ID	VarChar(20) NotNull
Tag	VarChar(20) NotNull
Name	Varchar(60) Not Null
Date	Datetime

	Not Null

Constraints:

- **Primary Key ➞** (ID, Tag, Name, Date)
- **Foreign Key ➞** (ID, Name, Date) references Contest(ID, Name, Date)
- **Check ➞** Tag in {graphs, DP, binary search, greedy, implementation, data structures, brute force, math, strings, number theory}

GROUP

Name	Varchar(50) NotNull
DateOfFormation	Date Not Null
Size	Int Default = 1 Not Null
Group_ID	VarChar(20) NotNull Unique
ID	VarChar(20) NotNull

Constraints:

- **Primary Key ➞** Group_ID
- **Foreign Key ➞** ID references USER(ID) (Is an admin)

RECRUITER

ID	VarChar(20) NotNull Unique
Email	VarChar(100) Unique NotNull
Name	VarChar(50)

	NotNull
Location_City	VarChar(50) NotNull
Location_Country	VarChar(50) NotNull
DateOfJoining	Date NotNull

Constraints:

- **Primary Key** → ID

ADMIN

Role	VarChar(50) NotNull Unique
Revenue	Float Default = 0
Email	VarChar(100) Unique NotNull

Constraints:

- **Primary Key** → Roles

PROGRAMMING_ORGANIZATION

ID	VarChar(20) NotNull Unique
Revenue Spent	float Default = 0
Name	VarChar(50) NotNull
Email	VarChar(100)

	Unique NotNull

Constraints:

- **Primary Key ➝** ID
- **Derived ➝** (% change in users)

RELATIONSHIP TABLES

MEMBER_OF

ID	VarChar(20) NotNull
Group_ID	VarChar(20) NotNull
DateOfJoining	Date NotNull

Constraints:

- **Primary Key ➝** (ID, Group_ID)
- **Foreign Key ➝** ID references USER(ID)
- **Foreign Key ➝** Group_ID references GROUP(Group_ID)

FOLLOWING

User_ID	VarChar(20) NotNull
Following_ID	VarChar(20) NotNull

Constraints:

- **Primary Key ➝** (User_ID, Following_ID)
- **Foreign Key ➝** User_ID references USER(ID)
- **Foreign Key ➝** Following_ID references USER(ID)

SOLVED

User_ID	VarChar(20) NotNull
Problem_ID	Int NotNull
Language	VarChar(20) NotNull
Date	Datetime NotNull

Constraints:

- **Primary Key ➞** (User_ID, Problem_ID, Language, Date)
- **Foreign Key ➞** User_ID references USER(ID)
- **Foreign Key ➞** Problem_ID references PROBLEMS(ID)
- **Check ➞** Language in {C++, C, Python, Java, JS, Kotlin, C#}

REGISTERED

User_ID	VarChar(20) NotNull
Programming_Organisation_ID	VarChar(20) NotNull
DateOfJoining	Date NotNull

Constraints:

- **Primary Key ➞** (User_ID, Programming_Organisation_ID)
- **Foreign Key ➞** User_ID references USER(ID)
- **Foreign Key ➞** Programming_Organisation_ID references PROGRAMMING ORGANISATION(ID)

RECRUITED

Recruiter_ID	VarChar(20) NotNull
User_ID	VarChar(20)

	NotNull
DateOfRecruitment	Date NotNull

Constraints:

- **Primary Key ➞** (Recruiter_ID, User_ID)
- **Foreign Key ➞** User_ID references USER(ID)
- **Foreign Key ➞** Recruiter_ID references RECRUITER(ID)

PREFERRED

Recruiter_ID	VarChar(20) NotNull
User_ID	VarChar(20) NotNull

Constraints:

- **Primary Key ➞** (User_ID, Recruiter_ID)
- **Foreign Key ➞** User_ID references USER(ID)
- **Foreign Key ➞** Recruiter_ID references RECRUITER(ID)

PREMIUM_PAYS_TO

Premium_User_ID	VarChar(20) NotNull

Admin_Role	VarChar(50)
	NotNull
	Default = “Admin_Premium”

DateOfPayment	DateTime NotNull
AmountPaid	Int NotNull

Constraints:

- **Primary Key ➞** (Premium_User_ID, Admin_Role, DateOfPayment)
- **Foreign Key ➞** Premium_User_ID references USER(ID)

- **Foreign Key ➞** Admin_Role references ADMIN(Role)

ORGANIZATION_PAYS_TO

Organization_ID	VarChar(20) NotNull

Admin_Role	VarChar(50)
	NotNull
	Default = “Admin_Organization”

DateOfPayment	DateTime NotNull
AmountPaid	Int NotNull
Ads	Text

Constraints:

- **Primary Key ➞** (Premium_User_ID, Admin_Role, DateOfPayment)
- **Foreign Key ➞** Premium_User_ID references USER(ID)
- **Foreign Key ➞** Admin_Role references ADMIN(Role)

TODOLIST

User_ID	VarChar(20) NotNull
Problem_ID	Int NotNull
Name	Varchar(60) Not Null

Constraints:

- **Primary Key ➞** (User_ID, Problem_ID, Name)
- **Foreign Key ➞** (User_ID, Name) references REPOSITORY(ID, Name)
- **Foreign Key ➞** Problem_ID references PROBLEM(ID)

FAVOURITES

User_ID	VarChar(20) NotNull
Problem_ID	Int NotNull
Name	Varchar(60) Not Null

Constraints:

- **Primary Key →** (User_ID, Problem_ID, Name)
- **Foreign Key →** (User_ID, Name) references REPOSITORY(ID, Name)
- **Foreign Key →** Problem_ID references PROBLEMS(ID)