Akshya Ramesh

AR Mobile App

This is a database design for an AR mobile-based application, is a friend you can look out for in a new city like Indianapolis. The main aim of the application is to help newcomers or residents with information around the city including but not limited to popular restaurants or building information with the latest tagged posts from Instagram, Facebook, and Twitter. The application is built using AR technology to popup information along with Unity to have a user-friendly application. To spice up personalized agents or build agents are added in familiar places. Each agent can carry a message and traverse through building assigned to them. The GPS (Global Positioning System) location of the users is constantly tracked when they use this application and popup information, they prefer to see in proximity which is previously stored along with building information. Each play can also collect agents from buildings to increase the interactions of the customers with application.

The aim of designing this database is to help the developers to store, retrieve, and update information pertaining to buildings, person location details, and avatars. There are going to be constant updates and retrievals that need to be done when the application is being used. Potential two main users for this database are the developers (admin) and players (users). Both developers and players will have access to the database, but players will only have limited access to a few of the tables. A robust database management system is designed to run the application using 13 tables. A detailed overview of each table with its usage and accessibility are mentioned below.

PERSON entity is used to store basic information about the users including Name, DOB, Age, Email, Username, Password, and Zipcode. A user will save this information when they first register to the application. A role attribute is used to differentiate the admin from other users. Admin will have access to perform DML operations to this table and users will have access to only create and update their own records the table.

LOGIN & LOGIN_HISTORY entities are self-explanatory. Login entity contains the username and password related to the person table. A new record is created in the login history table every time a user logs in. It stores information including Date, login time, logout time, IP Address, and login ID. Both admin and users have only create and read access to this table.

BUILDINGS entity stores information related to building along with Name, Description, GPS Coordinates, Type. Buildings tables is a super type and is related with one subtype entity – BUILDING TYPE. The subtype entity is used when there are more than one building types associated with a building. For e.g.: a Mall which is a super type can contain multiple sub entities like restaurants,

theater, shops etc. Each subtype contains information specific to type, address, room number, floor number, and detailed description. Admin has access to all the DML operations, whereas users can only read the records from this table.

PERSON_GPS is a composite entity between Users and Buildings. Records are created in this table when a user spends more than 1 min on a specific GPS location. Details related to time, date, user and building are recorded in this entity. Admin has access to all the DML operations, whereas users can only create new records to this table.

HASHTAG is an entity that records all the posts shared specific to the buildings on Instagram, Twitter, and Facebook. It stores the details about the users who shared the posts along with User Details, Type of post, Caption, image link, and Type of social media. Admin has access to all the DML operations, whereas users can only read the records from this table.

AGENT entity is used to store agent details related to Name, Description, GPS Coordinates. Admin has access to all the DML operations, whereas users can only read the records from this table.

TASK entity contains details related to the tasks assigned to each agent, Each task resord has a task ID, task name and task description. Admin has access to all the DML operations, whereas users can only read the records from this table.

AGENT_TASK entity stores tasks for each agent like task ID, Priority, Agent. Each agent task can be assigned to zero or many agent. Only admin has access to this entity with all the DML operations.

AGENT_PERSON entity is a composite entity where users can collect agents from any location. It stores Player ID, and Agent ID. Admin has access to all the DML operations, whereas users can only create new records to this table.

CONTENT entity is used to store contents or messages that each agent is assigned to. It can store all types of messages including the content ID, message, and type of message (social, advertising/marketing). Admin has access to all the DML operations, whereas users can only read records from this table.

AGENT_CONTENT entity is a composite entity that is used in assigning messages to the agents. Each agent can have more than one content assigned to a specific building. Admin has access to all the DML operations, whereas users can only read existing records to this table.

AGENT_BUILDING is used as a composite entity between buildings and agent content. Agents assigned with contents visiting multiple buildings are recorded in this entity. It stores the building

id, agent content id, and GPS coordinates. Admin has access to all the DML operations, whereas users can only create new records to this table.

DATA DICTIONARY:

Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK Referenced Table
PERSON	PER_ID	Contains person ID	CHAR(6)	P99999	P00001 - P99999	Y	PK	
	PER_FNAME	Contains person first name	VARCHAR(20)	Xxxxxx				
	PER_LNAME	Contains person last name	VARCHAR(20)	Xxxxxx				
	PER_DOB	Contains person Date of Birth	DATE	YYYY-MM-DD				
	PER_EMAIL	Contains person Email Id	VARCHAR(30)	xxx@xxx.xxx		Y		
	PER_ROLE	Contains the role of the user	CHAR(10)	Xxxxx				
	PER_ZIPCODE	Contains persons zipcode	CHAR(5)	99999				
LOGIN	LOG_USER_NAME	Contains the unique user name	VARCHAR(50)	XXXX		Y	PK	
	LOG_PASSWORD	Contains the password	VARCHAR(30)	XXXXX				
	PER_ID	Contains the reference to person table	CHAR(6)	P99999			FK	PERSON
LOGIN_HISTORY	HIS_ID	Contains login history ID	CHAR(10)	H999999999	H0000 00001 -	Y	PK	

					H9999 99999			
	HIS_DATE	Contains login Date	DATE	YYYY-MM-DD		Y		
	HIS_LOGIN_TIME	Contains login start time	TIME	HH:MM:SS		Y		
	HIS_LOGOUT_TIME	Contains logout time	TIME	HH:MM:SS				
	HIS_IPADDRESS	Contains the IP address of the logged in device	VARCHAR(15)	999.999. 9.99				
	LOGIN_USERNAME	Contains the username	VARCAR(30)	Xxxxxx		Y	FK	LOGIN
BUILDING	BUILD_ID	Contains building ID	CHAR(7)	B9999999	B000001 - B999999	Y	PK	
	BUILD_NAME	Contains building Name	VARCHAR(50)	Xxxxxx				
	BUILD_GPS_LATITUDE	Contains GPS latitude location	VARCHAR(20)	99999.999999999		Y		
	BUILD_GPS_LONGITUDE	Contains GPS longitude location	VARCHAR(20)	99999.999999999		Y		
	BUILD_DESCTRIPTION	Contains a short description about the building	VARCHAR(100)	Xxxx xxx xxx	`			
	BUILD_ADDRESS_LINE1	Contains Address	VARCHAR(50)	Xxxxxxxx				
	BUILD_CITY	Contains the city	VARCHAR(20)	Xxxxx				
	BUILD_ZIPCODE	Contains the zipcode	VARCHAR(5)	55555				

OFFICE_BUILDING	BUILD_ID	Contains building ID	CHAR(7)	9999999	0000001 - 9999999	Y	PK, FK	BUILDING
	OB_NO_OF_FLOOR	Contains details about the building	INT(3)	999				
	OB_BUILDING_STRENGTH	Contains details about the building	INT(3)	999				
	OB_OFFICE_TYPE	Contains details about the building	VARCHAR(50)	Xxxxx				
RESTURANT	BUILD_ID	Contains building ID	CHAR(7)	9999999	0000001 - 9999999	Y	PK, FK	BUILDING
	RES_GOOGLE_RATING	Contains details about google rating	FLOAT	9.9				
	RES_YELP_RATING	Contains details about yelp rating	FLOAT	9.9				
	RES_NO_OF_STAFF_WORKING	Contains details about the restaurant	INT(3)	999				
	RES_RESTURANT_TYPE	Contains details about the restaurant	VARCHAR(50)	XXXX				
MALL_BUILDING	BUILD_ID	Contains building ID	CHAR(7)	9999999	0000001 - 9999999	Y	PK, FK	BUILDING
	MALL_NO_OF_SHOPS	Contains details about google rating	INT(6)	9				

	MALL_NO_OF_RESTURANTS	Contains details about yelp rating	INT(6)	9				
	MALL_NO_OF_FLOOR	Contains details about the restaurant	INT(3)	999				
HASTAG	TAG_ID	Contains the hashtag ID	CHAR(7)	9999999	H000001-H999999	Y	PK	
	BUILD_ID	Contains the building ID	CHAR(7)	9999999	0000001 - 9999999		FK	BUILDING
	TAG_USERNAME	Contains the username of the post shared	VARCHAR(25)	Xxxxxx				
	TAG_TYPE	Specifies the type of post	CHAR(100)	Xxxxxx				
	TAG_PLATFORM	Specifies the platform in which it has been shared	CHAR(10)	Xxxxxx				
	TAG_LINK	Contains the link of the post	VARCHAR(300)	xxxxx.xxxx				
PERSON_GPS	PG_ID	Contains a unique ID	CHAR(5)	99999	00001-99999	Y	PK	
	BUILD_ID	Contains building ID	CHAR(7)	B999999	B000001 - B999999	Y	FK	BUILDING
	PER_ID	Contains the person ID	CHAR(6)	P99999	10000 - 999999	Y	FK	PERSON
	PG_DATE	Contains the date	DATE	YYYY-MM-DD				

	PG_TIME	Contains the time	TIME	hh:mm:ss				
AGENT	AG_ID	Contains agent ID	CHAR(5)	A9999	A0001-A9999	Y	PK	
	AG_NAME	Contains agent name	VARCHAR(25)	Xxxxx				
	AG_GPS_LATITUDE	Contains GPS Latitude	VARCHAR(20)	99999.999999999		Y		
	AG_GPS_LONGITUDE	Contains GPS Longitude	VARCHAR(20)	99999.999999999		Y		
	AG_DESC	Contains description of the agent	VARCHAR(50)	Xxxxx xxxx xxxx				
TASK	TASK_ID	Contains the task ID	CHAR(6)	T9999	T0001 - T9999	Y	PK	
	TASK_NAME	Contains the name of the task	VARCHAR(25)	Xxxxx				
	TASK_DESCRIPTION	Contains the description of the task	VARCHAR(100)	Xxxxx				
	TASK_TYPE	Contains the type of task	VARCHAR(25)	Xxxxx				
AGENT_TASK	AT_ID	Contains the agent ID	CHAR(6)	AT9999	AT0001 - AT9999	Y	PK	
	TASK_ID	Contains the task ID	CHAR(6)	T9999	T0001 - T9999	Y	FK	TASK
	AGENT_ID	Contains the agent ID	CHAR(5)	A9999	A0001-A9999	Y	FK	AGENT
	AT_PRIORITY	Priority of the task to a particular agent	NUMBER(2,0)	1 - 99				
AGENT_PERSON	AP_ID	Contains the unique ID	CHAR(8)	99999999	0000001-	Y	PK	

Akshya Ramesh

					999999 99			
	PER_ID	Contains person ID	CHAR(6)	P99999	10000 - 999999	Y	FK	PERSON
	AG_ID	Contains agent ID	CHAR(6)	AC9999	AC0001 - AC9999	Y	FK	AVATAR
	AP_DATE	Captures the date	DATE	YYYY-MM-DD				
	AP_TIME	Captures the time	TIME	hh:mm:ss				
CONTENT	CONTENT_ID	Contains the unique ID	CHAR(8)	C9999999	C0000001- C9999999	Y	PK	
	CONTENT_MESSAGE	Contains the message	VARCHAR(100)	Xxxxx		Y		
	CONTENT_TYPE	Contains the type of message	VARCHAR(50)	Xxxxxx				
AGENT_CONTENT	AC_ID	Contains the unique ID	CHAR(8)	99999999	CA000001- CA999999	Y	PK	
	CONTENT_ID	Contains the content ID	CHAR(8)	99999999		Y	FK	CONTENT
	AGENT_ID	Contains the agent ID	CHAR(5)	A9999		Y	FK	AGENT
BUILDING_AGENT	BA_ID	Contains the unique ID	CHAR(8)	99999999	BA000001- BA999999	Y	PK	
	AG_ID	Contains the agent content ID	CHAR(8)	99999999		Y	FK	AGENT_CONTENT
	BUILDING_ID	Contains the building ID	CHAR(7)	9999999		Y	FK	BUILDING

Akshya Ramesh

BUSINESS RULES:

- → Each person can have only one login; Each login is related to one and only one person
- → Each login has zero or many login history; One login history can have one and only one login
- → Building is an Office Building; Office Building is a Building
- → Building has a Restaurants; Restaurants is a Building
- → Building can be Mall Building; Mall Buildings is a Building
- → One building can have zero or many hashtags; One hashtag is related to zero or one building
- → Each task can be related to zero or many agents; Each agent can be assigned to zero or many tasks
- → Each person can be related to zero or many buildings; Each building can be related to zero or many person
- → Each agent can be caught by zero or many person; Each person can catch zero or many agents
- → Each content can be assigned to zero or many agents. Each agent can be assigned to zero or many contents.
- → Each agent content can be assigned to zero or many buildings; Each buildings can be assigned to zero or many agent content

ENTITY RELATIONSHIP MODEL (ERM) COMPONENTS:

Entity	Relationship	Connectivity	Entity
Person	has a	1:1	Login
Login	generates	1:M	Login History
Building	contains	1:M	Hashtag
Building	has	1:1	Office Building
Building	has	1:1	Restaurants
Building	Has	1:1	Mall Building
Building	viewed by	M:N	Person
Agent	Assigned	M:N	Task
Agent	caught by	M:N	Player
Content	Assigned to	M:N	Agent
Building	Assigned to	M:N	Agent Content

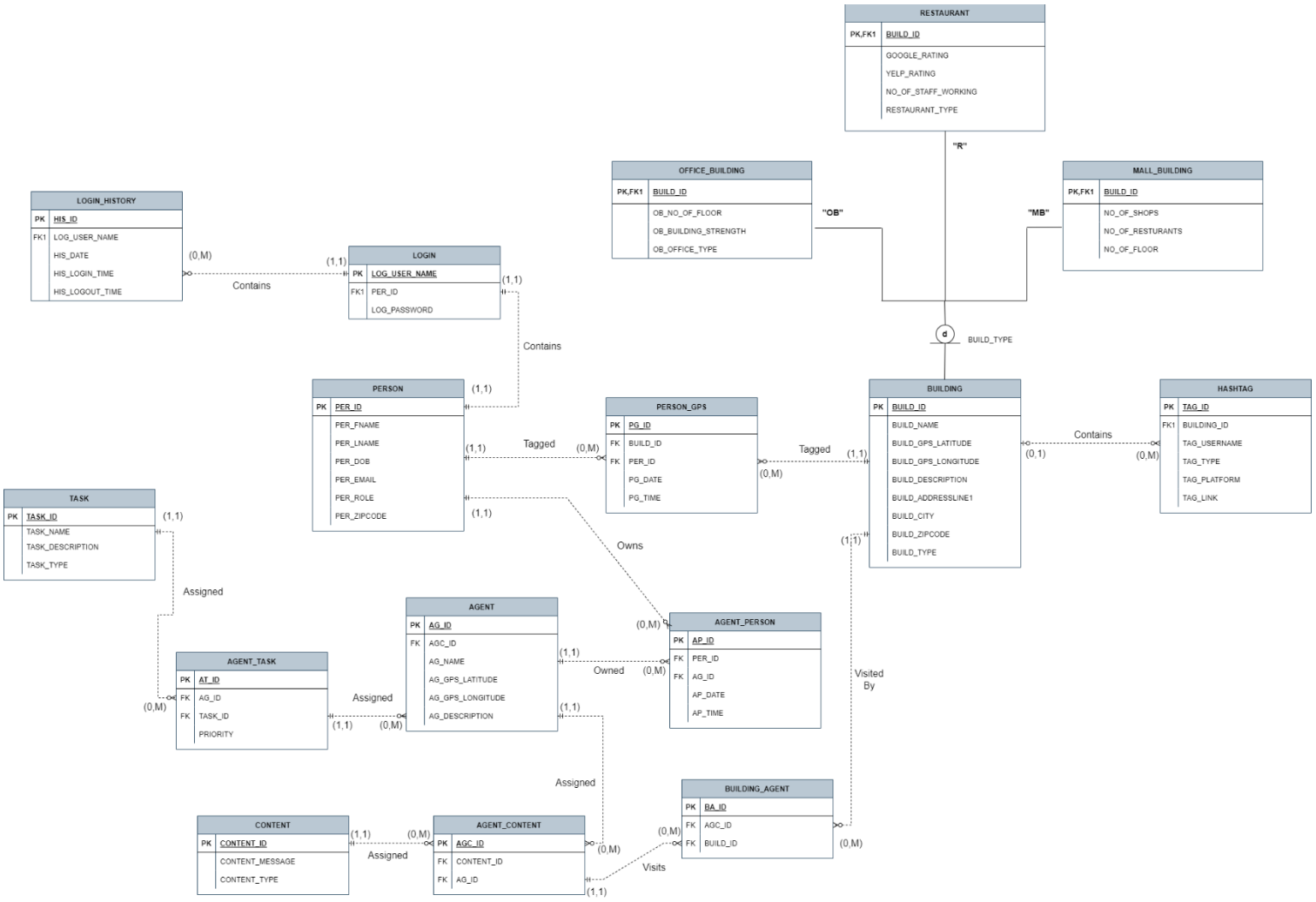
Akshya Ramesh

RELATION SCHEMA

- PERSON (**PER_ID**, PER_FNAME, PER_LNAME, PER_DOB, PER_EMAIL, PER_ROLE, PER_ZIPCODE)
- LOGIN(**LOG_USERNAME**, **PER_ID**,LOG_PASSWORD)
- LOGIN_HISTORY (**HIS_ID**, **LOG_USERNAME**, HIS_DATE, HIS_LOGIN_TIME, HIS_LOGOUT _TIME)
- BUILDING(**BUILD_ID**,BUILD_NAME,BUILD_GPS_LATITUDE,BUILD_GPS_LONGITUDE,BUILD_ DESCRIPTION,BUILD_ADDRESSLINE1, BUILD_CITY, BUILD_ZIPCODE, BUILD_TYPE)
- OFFICE_BUILDING (**BUILD_ID**, OB_NO_OF_FLOOR, OB_BUILDING_STRENGTH, OB_OFFICE_TYPE)
- MALL_BUILDING (**BUILD_ID**, NO_OF_SHOPS, NO_OF_RESTURANTS, NO_OF_FLOOR)
- RESTAURANT (**BUILD_ID,** GOOGLE_RATING, YELP_RATING, NO_OF_STAFF_WORKING, RESTAURANT_TYPE)
- HASTAG (**TAG_ID**, **BUILD_ID**, TAG_USERNAME, TAG_TYPE, TAG_PLATFORM, TAG_LINK)
- PERSON_GPS (**PG_ID**, **BUILD_ID**, **PER_ID**, PG_DATE, PG_TIME)
- AGENT (**AG_ID**, AG_NAME, AG_GPS_LATITUDE, AG_GPS_LONGITUDE, AG_DESC)
- TASK (**TASK_ID**, TASK_NAME , TASK_TYPE, TASK_DESCRIPTION)
- AGENT_TASK(**AT_ID, TASK_ID, AG_ID**, AT_PRIORITY)
- AGENT_PERSON (**AP_ID**, **PER_ID**, **AG_ID**, AP_DATE, AP_TIME)
- CONTENT (**CONTENT_ID**, CONTENT_MESSAGE, CONTENT_TYPE)
- AGENT_CONTENT (**AC_ID**, **CONTENT_ID**, **AG_ID**)
- BUILDING_AGENT (**BA_ID**, **AG_ID**, **BUILDING_ID**)

Akshya Ramesh

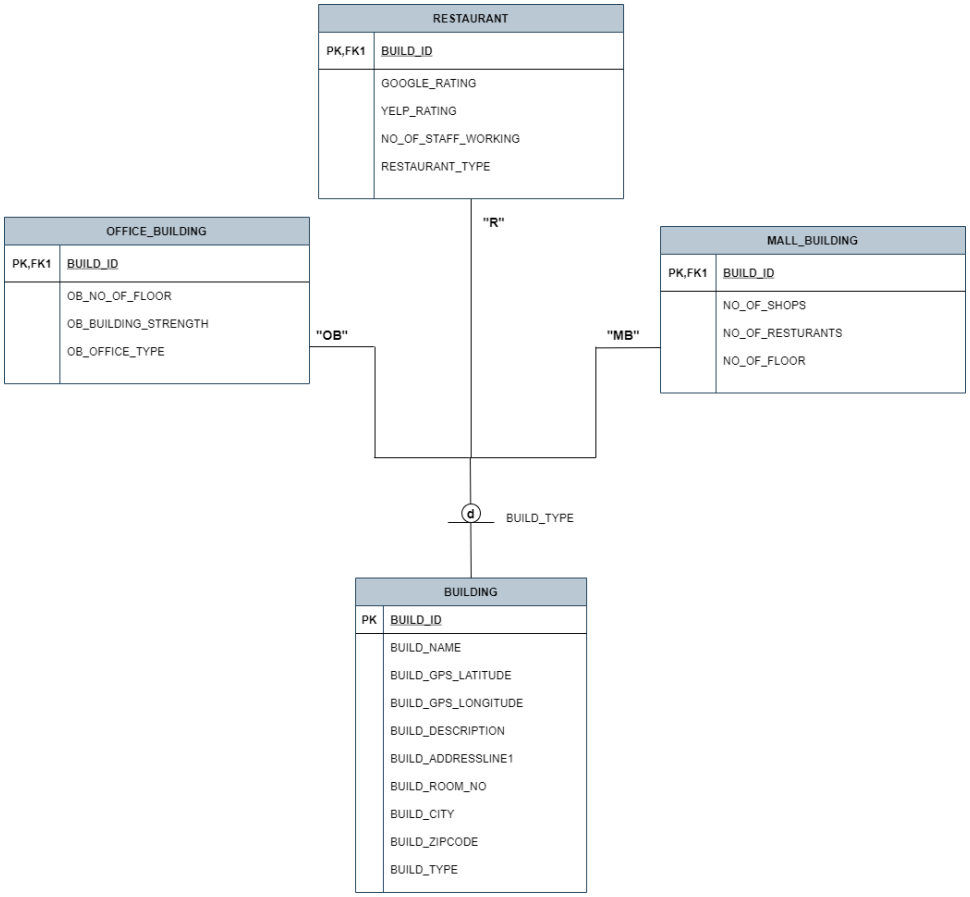
ER Diagram Crow's Foot Notation



Super and Sub Type

Building entity is a super type with has a disjoined partial relationship with three subtype entities, Office Building, Restaurants and Mall Building. These tables are linked with each other by Build Type attribute. The super entity Building contains all the common attributes like Building Name, GPS coordinates, Description, Address, and Building Type. Each Subtype contains attributes specific to each type.

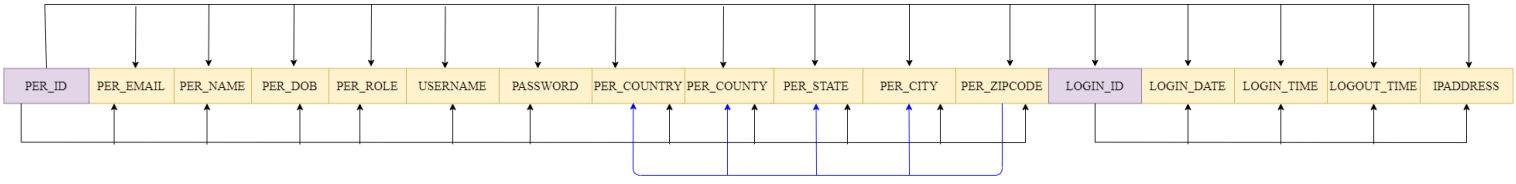
Akshya Ramesh



Normalization

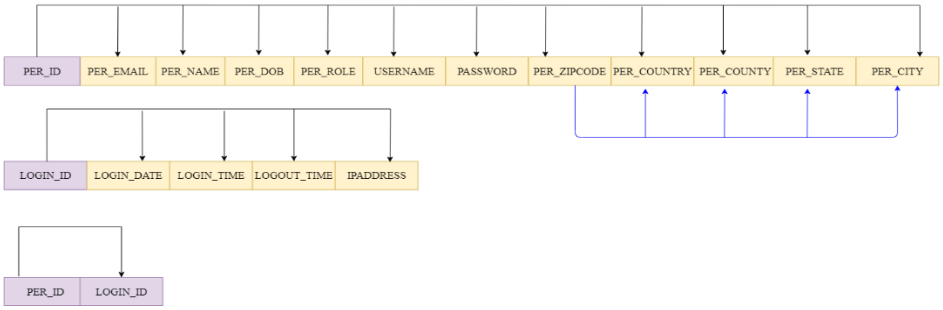
1NF

1NF contains both partial and transitive dependency. Partial dependencies are Login ID and Per ID. Transitive dependency is Zip code. Primary attributes are highlighted in purple and non-prime attributes are highlighted in yellow in the below image to distinguish. The black arrows in the bottom represent attributes that are partially dependencies associated with the primary key attributes. The blue arrows represent transitive dependencies.



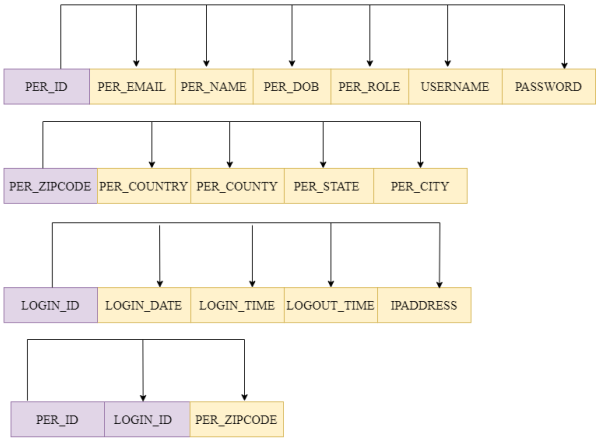
2NF

The main condition for 2NF is 1NF along with eliminating partial dependencies. In 2NF partial dependencies are removed and separate tables are created for corresponding dependent attributes. The current normalized form has 3 tables as follows: Person ID, Login History ID and a composite entity that stores the person ID and Login History ID.

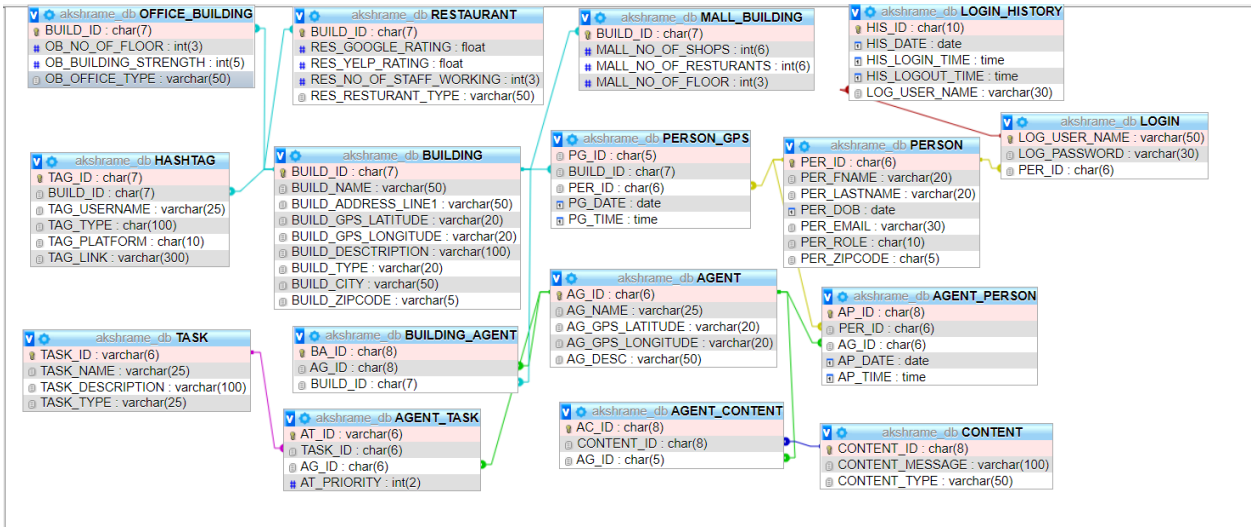


3NF

3NF has the condition of 2NF + removing transitive dependencies. This normalized table has 4 tables as follows: Person ID, zip code, Login History ID, and one composite entity that connects all the 3 tables (Person, zip code, and Login History). 3NF is the highest normalized tables as shown below.



Relationship



Working SQL queries

a. Fetch the details of person with birthday month in December. **[One Table]**

```
SELECT * FROM `PERSON` WHERE MONTH(PER_DOB) = 12
```

OUTPUT:

+ Options

	PER_ID	PER_FNAME	PER_LASTNAME	PER_DOB	PER_EMAIL	PER_ROLE	PER_ZIPCODE
Edit Copy Delete	P00005	Rams	Walk	2001-12-03	rames@gmail.com	Admin	45222

b. Get person details with user name that contains jame letters. **[SUB-QUERY]**

```
SELECT PER_ID, PER_FNAME, PER_LASTNAME, PER_DOB FROM PERSON WHERE
PER_ID IN (SELECT PER_ID FROM LOGIN WHERE LOG_USER_NAME LIKE
'%jame%')
```

OUTPUT:

+ Options

	PER_ID	PER_FNAME	PER_LASTNAME	PER_DOB
Edit Copy Delete	P00004	James	Mary	1990-05-03

c. Fetch login user details of users with login in history **[Two Tables]**

```
SELECT * FROM `LOGIN_HISTORY` AS LH INNER JOIN LOGIN AS L WHERE L.
LOG_USER_NAME = LH.LOG_USER_NAME
```

OUTPUT:

HIS_ID	HIS_DATE	HIS_LOGIN_TIME	HIS_LOGOUT_TIME	LOG_USER_NAME	LOG_USER_NAME	LOG_PASSWORD	PER_ID
H000000001	2021-06-12	40:23:11	50:23:11	david.will@gmail.com	david.will@gmail.com	qwerty12345	P00001
H000000002	2021-10-10	30:23:11	50:13:00	david.will@gmail.com	david.will@gmail.com	qwerty12345	P00001
H000000003	2020-01-10	10:20:11	30:23:11	rames@gmail.com	rames@gmail.com	something	P00005
H000000004	2021-11-05	30:20:11	50:23:11	steve@outlook.com	steve@outlook.com	whitecolor	P00002
H000000005	2021-08-04	30:20:11	50:23:11	sam@yahoo.com	sam@yahoo.com	manifest	P00003