**Create table:**

* create database social\_media;
* use social\_media;

CREATE TABLE `social\_media`.`aut\_sys` (

`user\_id` INT NOT NULL AUTO\_INCREMENT,

`user\_name` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`password` VARCHAR(45) NULL,

`status` VARCHAR(45) NULL,

`code` INT,

PRIMARY KEY (`user\_id`));

CREATE TABLE `social\_media`.`manager` (

`m\_id` INT NOT NULL AUTO\_INCREMENT,

`first\_name` VARCHAR(45) NULL,

`last\_name` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`phone` VARCHAR(45) NULL,

`street` VARCHAR(45) NULL,

`zip` VARCHAR(45) NULL

PRIMARY KEY (`m\_id`);

CREATE TABLE `social\_media`.`talent` (

`t\_id` INT NOT NULL AUTO\_INCREMENT,

`first\_name` VARCHAR(45) NULL,

`last\_name` VARCHAR(45) NULL,

`app\_id` INT NULL,

`app\_name` VARCHAR(45) NULL,

`m\_name` VARCHAR(45) NULL,

`m\_id` INT NULL,

`phone` INT NULL,

`email` VARCHAR(45) NULL,

`dob` DATE NULL,

`join\_date` DATE NULL,

`ranking` INT NULL,

`refered\_by` VARCHAR(45) NULL,

`salary` INT NULL,

`street` VARCHAR(45) NULL,

`zip` INT NULL,

PRIMARY KEY (`t\_id`));

CREATE TABLE `social\_media`.`pincode` (

`zip` INT NOT NULL ,

`city` VARCHAR(45) NULL,

`state` VARCHAR(45) NULL,

`country` VARCHAR(45) NULL,

PRIMARY KEY (`zip`));

CREATE TABLE `social\_media`.`dashboard` (

`d\_id` INT NOT NULL AUTO\_INCREMENT,

`a\_id` INT NULL,

`t\_id` INT NULL,

`m\_id` INT NULL,

`posts` INT NULL,

`likes` INT NULL,

`shares` INT NULL,

`saves` INT NULL,

`link` VARCHAR(45) NULL,

PRIMARY KEY (`d\_id`));

CREATE TABLE `social\_media`.`agency` (

`a\_id` INT NOT NULL AUTO\_INCREMENT,

`a\_name` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`phone` INT NULL,

`street` VARCHAR(45) NULL,

`zip` INT NULL,

PRIMARY KEY (`a\_id`));

CREATE TABLE `social\_media`.`application` (

`app\_id` INT NOT NULL AUTO\_INCREMENT,

`app\_name` VARCHAR(45) NULL,

`no\_of\_talents` VARCHAR(45) NULL,

`app\_rank` VARCHAR(45) NULL,

PRIMARY KEY (`app\_id`));

CREATE TABLE `social\_media`.`recruits` (

`r\_id` INT NOT NULL AUTO\_INCREMENT,

`contract` INT NULL,

`t\_id` INT NULL,

`app\_id` INT NULL,

`add\_id` INTNS

PRIMARY KEY (`r\_id`));

CREATE TABLE `social\_media`.`log` (

`log\_id` INT NOT NULL AUTO\_INCREMENT,

`roles` VARCHAR(45) NULL,

`login\_date` DATE NULL,

`login\_time` TIME NULL,

`logout\_time` TIME NULL,

`user\_id` INT NULL,

PRIMARY KEY (`log\_id`));

CREATE TABLE `social\_media`.`reason` (

`code` INT NOT NULL,

`details` VARCHAR(100) NULL);

CREATE TABLE `social\_media`.`payment` (

`invoice\_id` INT NOT NULL AUTO\_INCREMENT,

`t\_id` INT NULL,

`amount` INT NULL,

`mode\_of\_payment` VARCHAR(45) NULL,

`pay\_date` DATE NULL,

PRIMARY KEY (`invoice\_id`));

CREATE TABLE `social\_media`.`posts` (

`p\_id` INT NOT NULL AUTO\_INCREMENT,

`upload\_date` DATE NULL,

`upload\_time` TIME NULL,

`type` VARCHAR(45) NULL,

`p\_link` VARCHAR(45) NULL,

`t\_id` INT NULL,

`m\_id` INT NULL,

PRIMARY KEY (`p\_id`));

CREATE TABLE `social\_media`.`advertisement` (

`add\_id` INT NOT NULL AUTO\_INCREMENT,

`app\_id` INT NULL,

`m\_id` INT NULL,

`t\_id` INT NULL,

`brand\_name` VARCHAR(45) NULL,

`partnership` VARCHAR(45) NULL,

`charges` INT,

PRIMARY KEY (`add\_id`));

CREATE TABLE `social\_media`.`report` (

`r\_id` INT NOT NULL AUTO\_INCREMENT,

`t\_id` INT NULL,

`m\_id` INT NULL,

`app\_id` INT NULL,

`rank` INT NULL,

`status` VARCHAR(45) NULL,

`no\_of\_uploads` VARCHAR(45) NULL,

`join\_date` VARCHAR(45) NULL,

PRIMARY KEY (`r\_id`));

CREATE TABLE `social\_media`.`deliverables` (

`promotion\_id` INT NOT NULL AUTO\_INCREMENT,

`t\_id` INT NULL,

`m\_id` INT NULL,

`app\_id` INT NULL,

`rank` INT NULL,

`status` VARCHAR(45) NULL,

`no\_of\_uploads` INT NULL,

`to\_be\_uploaded` INT NULL,

`add\_id` INT NULL,

`start\_data` DATE,

`end\_date` DATE );

CREATE TABLE `social\_media`.`location` (

`l\_id` INT NOT NULL AUTO\_INCREMENT,

`ip\_address` VARCHAR(45) NULL,

`region` VARCHAR(45) NULL,

PRIMARY KEY (`l\_id`));

CREATE TABLE `social\_media`.`display` (

`contact\_details` VARCHAR(25) NULL,

`email` VARCHAR(45) NULL,

`description` VARCHAR(200) NULL);

CREATE TABLE `social\_media. audit` (

`first\_name` VARCHAR(45) NULL,

`last\_name` VARCHAR(45) NULL,

`app\_id` INT NULL,

`app\_name` VARCHAR(45) NULL,

`m\_name` VARCHAR(45) NULL,

`m\_id` INT NULL,

`phone` INT NULL,

`email` VARCHAR(45) NULL,

`dob` DATE NULL,

`join\_date` DATE NULL,

`ranking` INT NULL,

`refered\_by` VARCHAR(45) NULL,

`salary` INT NULL,

`street` VARCHAR(45) NULL,

`zip` INT NULL

)

CREATE TABLE `social\_media. enquiry` (

`first\_name` VARCHAR(45) NULL,

`last\_name` VARCHAR(45) NULL,

`phone` INT NULL,

`email` VARCHAR(45) NULL,

`dob` DATE NULL,

`refered\_by` VARCHAR(45) NULL,

`Comments` VARCHAR(45),

`street` VARCHAR(45) NULL,

`zip` INT NULL

)

* alter table talent add foreign key(m\_id) references manager(m\_id);
* alter table agency add foreign key(zip) references pincode(zip);
* alter table manger add foreign key(zip) references pincode(zip);
* alter table talent add foreign key(zip) references pincode(zip);
* alter table media add foreign key(m\_id) references manager(m\_id);
* alter table media add foreign key(t\_id) references talent(t\_id);
* alter table recruits add foreign key(t\_id) references talent(t\_id);
* alter table recruits add foreign key(app\_id) references media(app\_id);
* alter table authentication\_system add foreign key(code) references reason(code);
* alter table log add foreign key(user\_id) references authentication\_system(user\_id);
* alter table payment add foreign key(t\_id) references talent(t\_id);
* alter table posts add foreign key(t\_id) references talent(t\_id);
* alter table posts add foreign key(m\_id) references manager(m\_id);
* alter table dashboard add foreign key(m\_id) references manager(m\_id);
* alter table dashboard add foreign key(t\_id) references talent(t\_id);
* alter table dashboard add foreign key(p\_id) references posts(p\_id);
* alter table advertisement add foreign key(app\_id) references media(app\_id);
* alter table advertisement add foreign key(t\_id) references talent(t\_id);
* alter table advertisement add foreign key(m\_id) references manager(m\_id);
* alter table reports add foreign key(t\_id) references talent(t\_id);
* alter table reports add foreign key(m\_id) references manager(m\_id);
* alter table reports add foreign key(app\_id) references media(app\_id);
* alter table deliverables add foreign key(p\_id) references posts(p\_id);
* alter table deliverables add foreign key(t\_id) references talent(t\_id);
* alter table deliverables add foreign key(m\_id) references manager(m\_id);
* alter table deliverables add foreign key(app\_id) references media(app\_id);

**SQL for insertion of data:**

INSERT INTO `social\_media`.`aut\_sys` (`user\_name`, `email`, `password`) VALUES ('AnithaShivaramaiah', '1234567A', 'anithashiva@gmail.com');

INSERT INTO `social\_media`.`manager` (`first\_name`, `last\_name`, `email`, `phone`, `street`, `zip`) VALUES ('Anitha', 'Shivaramaiah', 'anithashivaramaiah@gmail.com', '4692585497', '4859, Cedar springs Road', '76219');

INSERT INTO `social\_media`.`pincode` (`zip`, `city`, `state`, `country`) VALUES ('76219', 'Denton', 'Texas', 'USA');

INSERT INTO `social\_media`.`talent` (`first\_name`, `last\_name`, `app\_id`, `app\_name`, `m\_name`, `m\_id`, `phone`, `email`, `dob`, `join\_date`, `ranking`, `refered\_by`, `salary`, `street`) VALUES ('sowmya', 'sk', '002', 'instagram', 'anitha', '001', '23456786', 'sowmya.sk@gmail.com', '18/02/1994', '20/10/2021', '1', 'Shilpa', '100', '4859, Cedar Springs Road');

INSERT INTO `social\_media`.`talent` (`first\_name`, `last\_name`, `app\_id`, `app\_name`, `m\_name`, `m\_id`, `phone`, `email`, `dob`, `join\_date`, `ranking`, `refered\_by`, `salary`, `street`, `zip`) VALUES ('sowmya', 'sk', '002', 'instagram', 'anitha', '001', '23456786', 'sowmya.sk@gmail.com', '1994-12-22', ' ', '1', 'Shilpa', '100', '4859, Cedar Springs Road', '76219');

INSERT INTO `social\_media`.`application` (`app\_id`, `app\_name`, `no\_of\_talents`, `app\_rank`) VALUES ('1', 'Facebook', '1', '2');

INSERT INTO `social\_media`.`agency` (`a\_name`, `email`, `phone`, `street`, `zip`) VALUES ('Blue Agency', 'blue.agency@gmail.com', '253678936', '8056, Avenue A', '76219');

INSERT INTO `social\_media`.`log` (`log\_id`, `roles`, `login\_date`, `login\_time`, `logout\_time`) VALUES ('1001', 'MN', '2021-11-21', '11:45', '13:05');

INSERT INTO `social\_media`.`location` (`ip\_address`, `region`, `log\_id`) VALUES ('10:20:90:56', 'Dallas', '1002');

INSERT INTO `social\_media`.`payment` (`invoice\_id`, `t\_id`, `amount`, `mode\_of\_payment`) VALUES ('10001', '1', '100', 'cheque');

INSERT INTO `social\_media`.`dashboard` (`d\_id`, `app\_id`, `t\_id`, `m\_id`, `posts`, `likes`, `shares`, `saves`, `link`) VALUES ('10001', '1', '2', '7', '10', '2500', '150', '578', 'www.facebook/anitha/4567');

INSERT INTO `social\_media`.`deliverables` (`t\_id`, `m\_id`, `app\_id`, `rank`, `status`, `no\_of\_uploads`, `to\_be\_uploaded`, `add\_id`, `start\_date`, `end\_date`) VALUES ('3', '8', '4', '4', 'on-going', '10', '25', '002', '2021-11-10', '2021-11-20');

INSERT INTO `social\_media`.`posts` (`p\_id`, `upload\_date`, `upload\_time`, `type`, `p\_link`, `t\_id`, `m\_id`) VALUES ('12001', '2021-11-10', '10:30', 'Video', 'www.insrafeam.anitha123/', '4', '8');

INSERT INTO `social\_media`.`report` (`r\_id`, `t\_id`, `m\_id`, `app\_id`, `rank`, `status`, `no\_of\_uploads`) VALUES ('1001', '1', '8', '2', '2', 'Active', '35');

INSERT INTO `social\_media`.`advertisement` (`add\_id`, `app\_id`, `m\_id`, `t\_id`, `brand\_name`, `charges`) VALUES ('10001', '4', '8', '3', 'zara', ‘75’

);

**SQLs for fetching data from different tables:**

1. **Initially when the first page is seen, and a Manager wants to add his account to login and enters all the data like, user\_name, password, email we use the below SQL:**

INSERT INTO `social\_media`.`aut\_sys` (`user\_name`, `email`, `password`, `status`) VALUES ('AnithaShivaramaiah', '1234567A', 'anithashiva@gmail.com', `MN`);

Here, to differentiate between the manager account and the talent account, we hardcode the values ‘MN’ for manager and then insert the data

1. **When the first page is seen, and a Talent wants to add his account to login and enters all the data like, user\_name, password, email we use the below SQL:**

INSERT INTO `social\_media`.`aut\_sys` (`user\_name`, `email`, `password`,`status`) VALUES ('AnithaShivaramaiah', '1234567A', 'anithashiva@gmail.com', `TL`);

Here, to differentiate between the manager account and the talent account, we hardcode the values ‘TL’ for talent and then insert the data.

1. **To check if the account already exists, during registering account we can use:**

SELECT \* FROM aut\_sys WHERE user\_name = %s AND password = %s AND status = %s' , (user\_name, password, status)

Where, the user\_name, password and the status with the access specifier %s is the data which the user enters on the UI while registering.

* If the entered data is already present in the aut\_sys table, we get a message telling the account already exist.

1. **Once after registering, we try to login, and we can use the following SQL to check if the correct credentials are entered or not:**

SELECT \* FROM aut\_sys WHERE user\_name = %s AND password = %s AND status = %s' , (user\_name, password, status)

Where, the user\_name, password and the status with the access specifier %s is the data which the user enters on the UI while logging in.

* If the entered data is present and an incorrect password is entered, then we get a message telling, “the entered password is incorrect”.

**FOR MANAGER ACCOUNT:**

1. **After we login with the correct credentials, to display the initial list of talents under a particular manager we can use:**

SELECT t\_id,first\_name, last\_name, dob, app\_name,join\_date, ranking, m\_name, email, phone FROM talent' WHERE m\_id = %s ', (m\_id)

1. **To display the detailed information of talent under a particular manager we can use:**

SELECT t\_id, first\_name, last\_name, dob, app\_id, app\_name, join\_date, ranking, email, phone, refered\_by, salary, street, zip FROM talent WHERE m\_id = %s ', (m\_id)

* This query we can get the talent details which is related to only the manager who has logged in.

1. **While displaying all the talent information on the screen, with the help of Zip we can fetch all the address information from another table:**

SELECT city, state, country FROM pincode WHERE zip = %s, (zip)

* Where, the %s value for zip is the value from the talent table and the zip value is from the pincode table.

1. **Combining these two SQLs we can use:**

SELECT t\_id, first\_name, last\_name, dob, app\_id, app\_name, join\_date, ranking, email, phone, refered\_by, salary, street, zip, city, state, country FROM talent as t, pincode as p WHERE t.zip = p.zip and m\_id = (m\_id value which is passed as parameter from the login screen).

1. **To Add a talent under a particular manager, we can use:**

INSERT INTO talent VALUES (NULL, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)', (first\_name, last\_name, email,app\_id, app\_name,m\_id, m\_name ,phone,street,zip\_code, dob,join\_date, ranking,ref\_by, salary,)

Where, we can hardcode the manager value from the previous screen data, which is passed as parameter.

1. **To edit a particular talent data we can use:**

First, we need to fetch the talent id, from the previous screen, hence that value will be passed as a parameter to the edit screen and then the data can be updated

UPDATE talent SET first\_name= %s ,last\_name= %s ,dob = %s ,app\_name = %s , join\_date = %s ,ranking= %s ,email= %s ,phone= %s ,ref\_by= %s ,salary= %s ,street= %s ,zip= %s WHERE t\_id= %s, (first\_name, last\_name, dob, app\_name, join\_date, ranking, email, phone, ref\_by, salary, street, zip, t\_id)

All the access specifiers are the values which are entered by the manager while editing the talent details, all these data are fetched from the UI screen.

1. **Store the deleted record as a back up, to maintain the data, we can store in the audit table before delete statement is executed:**

INSERT INTO social\_media.audit (`first\_name`, `last\_name`, `app\_id`, `app\_name`, `m\_name`, `m\_id`, `phone`, `email`, `dob`, `join\_date`, `ranking`, `refered\_by`, `salary`, `street`) VALUES ( SELECT \* FROM social\_media.talent WHERE t\_id = t\_id)

To fetch the manager name field we can get it from the manager table, to get the application name we can get it from the application table using the below queries.

SELECT m\_name FROM social\_media.manager WHERE m\_id = %s, m\_id(this is the value from the screen passed as a parameter or it can be the value from the talent table)

SELECT app\_name FROM social\_media.applicaiton WHERE app\_id = %s, app\_id(this is the value from the screen passed as a parameter or it can be the value from the application table)

1. **To delete a particular talent data we can use:**

DELET FROM talent WHERE t\_id = %s, t\_id

Here, the t\_id value which is obtained as a parameter from the previous detailed screen, is used for deleting the record from the table.

1. **To display all the advertisement related to a particular talent under a particular manager we can use:**

SELECT t\_id, app\_id, app\_name, brand\_name, partnership, charges FROM advertisement where t\_id = %s and m\_id = %s, (t\_id, m\_id))

To fetch the remaining details regarding the manager name, the app name we can fetch it from the Manager table and Applications table with the below SQL:

SELECT m\_name FROM social\_media.manager WHERE m\_id = %s, m\_id(this is the value from the screen passed as a parameter or it can be the value from the talent table)

SELECT app\_name FROM social\_media.applicaiton WHERE app\_id = %s, app\_id(this is the value from the screen passed as a parameter or it can be the value from the application table)

1. **Combining all the above SQLs we can use one SQL:**

SELECT t\_id, app\_id, app\_name, brand\_name, partnership, charges, m\_name, app\_name FROM talent as t, manager as m, application as a WHERE t.t\_id = m.id and t.t\_id = a.t\_id and where m\_id = (m\_id value hardcoded from the screen value) and a\_id = (a\_id value hardcoded from the screen value)

1. **To Add the advertisement related to a particular talent under a particular manager we can use:**

INSERT INTO advertisement VALUES (NULL, %s, %s, %s, %s, %s, %s, %s)', (app\_id, m\_id, t\_id, brand\_name, partnership,charges,app\_name,)

Where, the app\_id, m\_id and the t\_id is all hardcoded from the previous page from the UI and is inserted into the table.

1. **To update the advertisement related to a particular talent under a particular manager we can use:**

UPDATE advertisement SET app\_id = %s ,app\_name= %s ,brand\_name= %s ,partnership= %s ,charges= %s WHERE t\_id= %s, (app\_id, app\_name, brand\_name, partnership, charges, t\_id)

1. **If the manager updates the application details on the screen, it should be updated in both the talent table as well as the application table:**

UPDATE talent SET first\_name= %s ,last\_name= %s ,dob = %s ,app\_name = %s , join\_date = %s ,ranking= %s ,email= %s ,phone= %s ,ref\_by= %s ,salary= %s ,street= %s ,zip= %s WHERE t\_id= %s, (first\_name, last\_name, dob, app\_name, join\_date, ranking, email, phone, ref\_by, salary, street, zip, t\_id)

All the access specifiers are the values which are entered by the manager while editing the talent details, all these data are fetched from the UI screen.

UPDATE application SET app\_name = %s WHERE app\_id= %s, ( app\_name, app\_id)

1. **To display all the dashboard related to a particular talent under a particular manager we can use:**

SELECT d\_id, app\_id, app\_name, m\_id, posts, likes, shares, saves, link FROM dashboard where t\_id = %s' and m\_id = %s, (t\_id, m\_id))

To fetch the remaining details regarding the manager name, the app name we can fetch it from the Manager table and Applications table with the below SQL:

SELECT m\_name FROM social\_media.manager WHERE m\_id = %s, m\_id(this is the value from the screen passed as a parameter or it can be the value from the talent table)

SELECT app\_name FROM social\_media.applicaiton WHERE app\_id = %s, app\_id(this is the value from the screen passed as a parameter or it can be the value from the application table)

1. **Combining all the above SQLs we can use one SQL:**

SELECT d\_id, app\_id, app\_name, m\_id, posts, likes, shares, saves, link FROM dashboard as d, manager as m, application as a WHERE d.t\_id = m.id and d.t\_id = a.t\_id and where m\_id = (m\_id value hardcoded from the screen value) and a\_id = (a\_id value hardcoded from the screen value)

1. **To Add the dashboard related to a particular talent under a particular manager we can use:**

INSERT INTO dashboard VALUES (NULL, %s, %s, %s, %s, %s, %s, %s, %s, %s)', (app\_id, t\_id, m\_id, posts, likes, shares, saves, link, app\_name)

Where, the app\_id, m\_id and the t\_id is all hardcoded from the previous page from the UI and is inserted into the table.

1. **To update the dashboard information related to a particular talent under a particular manager we can use:**

UPDATE dashboard SET app\_id = %s , posts = %s ,likes = %s ,shares= %s ,saves= %s, links = %s, app\_name = %s WHERE t\_id= %s, and m\_id = %s (app\_id, posts, likes, shares, saves, links, app\_name, t\_id, m\_id)

1. **If the manager updates the application details on the screen, it should be updated in both the talent table as well as the application table:**

UPDATE application SET app\_name = %s WHERE app\_id= %s, ( app\_name, app\_id)

1. **To delete a particular advertisement details we can use the following:**

DELETE FROM talent WHERE d\_id = %s, d\_id(this value is the dashboard\_id which is hardcoded and passed as a parameter.

1. **To display all the promotions details for a particular advertisement we can use the following:**

SELECT promotion\_id, d\_id, app\_id, rank, status, no\_of\_uploads, to\_be\_uploaded, start\_date, end\_date FROM deliverables where t\_id = %s' and m\_id = %s and add\_id = %s, (t\_id,m\_id, add\_id))

Where the access specifiers are the data passed as the input from the UI screen.

1. **To Add the promotion related to a particular talent under a particular manager we can use:**

SELECT app\_id, m\_id, t\_id FROM social\_media. advertisement where add\_id = %s, (add\_id which is obtained as a parameter from the UI)

Once these values are fetched, it’s inserted into deliverables table.

INSERT INTO deliverables VALUES (NULL, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)', (t\_id, d\_id, app\_id, rank, status, no\_of\_uploads, to\_be\_uploaded, add\_id, start\_date, end\_date)

1. **To update the promotion details we can use the following:**

UPDATE deliverables SET rank = %s , status = %s , no\_of\_uploads = %s ,to\_be\_uploaded= %s ,add\_id= %s, start\_date = %s, end\_date = %s WHERE t\_id= %s, and m\_id = %s and add\_id = %s(rank, status, no\_of\_uploads, to\_be\_uploaded, add\_id, start\_date, end\_date, t\_id, m\_id, add\_id)

1. **To delete a particular promotion details we can use the following:**

DELETE FROM deliverables WHERE promotion\_id = %s and d\_id = %s, (these values are the promotion id and the dashboard\_id which is hardcoded and passed as a parameter from the previous screen from the UI)

1. **Manager generating the report for the talents based on the performance, we can use the following:**

INSERT INTO social\_media.report VALUES (NULL, %s, %s, %s, %s, %s, %s, %s), (t\_id, m\_id, app\_id, rank, status, no\_of\_uploads, join\_date)

1. **To display all the post information of a particular talent we can use:**

SELECT upload\_date, upload\_time, type, p\_link, t\_id, m\_id, app\_id FROM posts WHERE t\_id= %s and m\_id = %s, (t\_id, m\_id)

Where, m\_id and the t\_id are the talent and the manager information from the UI.

1. **To insert the post information of a particular talent we can use:**

INSERT INTO posts (NULL, %s, %s, %s, %s, %s, %s, %s) VALUES (upload\_time, upload\_date, type, p\_link, t\_id, m\_id, app\_id)

m\_id, app\_id can be obtained from the talent table using:

SELECT m\_id, app\_id FROM talent WHERE t\_id = %s (talent id from the UI)

1. **To update the post information of a particular talent we can use:**

UPDATE posts SET upload\_time = %s, upload\_date = %s, type = %s, p\_link = %s WHERE t\_id = %s, m\_id = %s, (upload\_time, upload\_date, type, p\_link, t\_id, m\_id all the data which the user enters on the screen in the UI.

1. **To obtain a payment information of a particular talent we can use :**

SELECT t\_id, amount, mode\_of\_payment, pay\_date FROM payment where t\_id = %s (t\_id)

1. **To Add a payment for a particular talent after a payment is made we can use:**

INSERT INTO payment (NULL, %s, %s, %s, %s) VALUES (t\_id, amount, mode\_of\_payment, pay\_date)

1. **To update a payment information:**

UPDATE payment SET amount = %s, mode\_of\_payment = %s, pay\_date = %s WHERE invoice\_id = %s and t\_id = %s, (invoice\_id, t\_id)

**FOR TALENT ACCOUNT:**

1. **If a user wants to enroll into a particular agency, they can enter details on the screen and those details will be entered into the enquiry table:**

INSERT INTO social\_media.enquiry VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s), (first\_name, last\_name, phone, email, dob, refered\_by, applications,comments, street, zip)

1. **After we login with the correct credentials, to display the initial list of talents under a particular talent we can use:**

SELECT t\_id,first\_name, last\_name, app\_id, app\_name, m\_name, m\_id, phone, email, dob, join\_date, ranking, refered\_by, salary, street, zip FROM talent WHERE t\_id = %s ', (t\_id)

* This query we can get the talent details which is related to only the talent who has logged in.

1. **While displaying all the talent information on the screen, with the help of Zip we can fetch all the address information from another table:**

SELECT city, state, country FROM pincode WHERE zip = %s, (zip)

* Where, the %s value for zip is the value from the talent table and the zip value is from the pincode table.

1. **Combining these two SQLs we can use:**

SELECT t\_id, first\_name, last\_name, dob, app\_id, app\_name, join\_date, ranking, email, phone, refered\_by, salary, street, zip, city, state, country FROM talent as t, pincode as p WHERE t.zip = p.zip and m\_id = (m\_id value which is passed as parameter from the login screen).

1. **For a talent to check all the information about the manager we can use the following query:**

SELECT first\_name, last\_name, email, phone, street, zip FROM manager WHERE m\_id = %s, (m\_id from the UI)

1. **For a talent to check the Promotions allotted to them we can use:**

SELECT app\_id, app\_name, brand\_name, partnership, charges FROM advertisement where t\_id = %s, (t\_id))

1. **For a talent to check the deliverables allotted to them we can use:**

SELECT app\_id, rank, status, no\_of\_uploads, to\_be\_uploaded, start\_date, end\_date FROM deliverables where t\_id = %s' and add\_id = %s, (t\_id, add\_id))

1. **For a talent to check the Dashboard we can use:**

SELECT app\_name, m\_id, posts, likes, shares, saves, link FROM dashboard where t\_id = %s' and m\_id = %s, (t\_id, m\_id))

1. **Initially to display the Agency information in the home screen, the data is obtained from:**

SELECT a\_name, email, phone, street, zip FROM agency where a\_id = %s,(a\_id which has different branch details)

1. **Each time a Manager or a Talent logs in or logs out of the system, a log entry is made on a daily basis to keep track it.**

To add the details into the log table, we use :

INSERT INTO log (NULL, %s, %s, %s, %s, %s) VALUES (roles, login\_date, login\_time, logout\_time, user\_id)

Where, the roles is hardcoded based on the talent or manager codes i.e `MN` or `TL`

1. **To display all the post information of a particular talent we can use:**

SELECT upload\_date, upload\_time, type, p\_link, t\_id, m\_id, app\_id FROM posts WHERE t\_id= %s (t\_id)

Where, t\_id is the talent information from the UI.

1. **To display the log details:**

SELECT log\_id, roles, login\_date, login\_time, logout\_time, user\_id FROM log

1. **To obtain a particular location details for the log details we can use:**

SELECT ip\_address, region FROM location WHERE log\_id = %s, (log\_id which is obtained from the log table)

1. **To display the details of the company on the screen of the home page we can use:**

SELECT contact\_details, email, description FROM social\_media.display