# ASSIGNMENT 4:

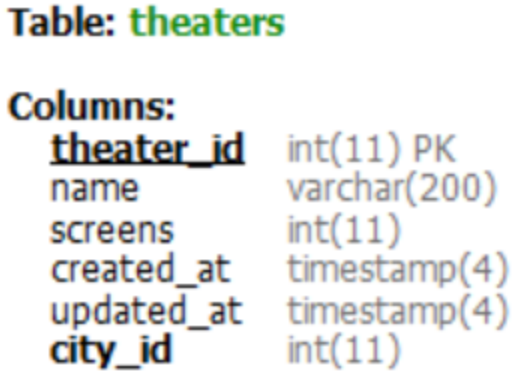
## TABLE THEATERS:

**Table ‘Theaters’ had an extra column ‘screens’ as we forgot to drop it earlier while creating the database as we already have a separate table ‘Screens’ which has ‘theater_id’ and ‘screen_id’ from where we can calculate the number of screens of the ‘Theaters’.**

**THEATERS TABLE BEFORE 3NF:**

```
CREATE TABLE `theaters` (
  `theater_id` int(11) NOT NULL,
  `name` varchar(200) DEFAULT NULL,
  `screens` int(11) DEFAULT NULL,
  `created_at` timestamp(4) NULL DEFAULT NULL,
  `updated_at` timestamp(4) NULL DEFAULT NULL,
  `city_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`theater_id`),
  KEY `city_id_fk_idx` (`city_id`),
  CONSTRAINT `city_id_fk` FOREIGN KEY (`city_id`) REFERENCES `city` (`city_id`) ON
DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `screens` (
  `screen_id` int(11) NOT NULL,
  `capacity` int(11) DEFAULT NULL,
  `theater_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`screen_id`),
  UNIQUE KEY `idScreens_UNIQUE` (`screen_id`),
  KEY `theater_id_sc_idx` (`theater_id`),
  CONSTRAINT `theater_id_sc` FOREIGN KEY (`theater_id`) REFERENCES `theaters`
(`theater_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```





**THEATERS TABLE AFTER 3NF:**

```
ALTER TABLE theaters
DROP COLUMN screens;
```

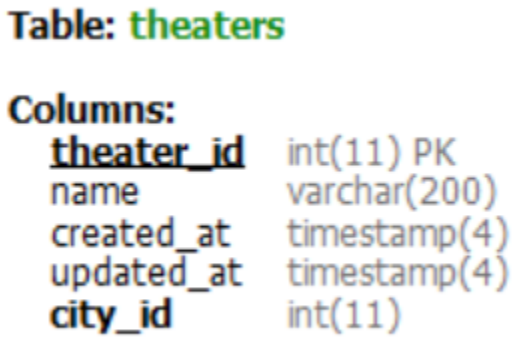



**TABLE USER:**

**Table ‘User’ had a Transitive Dependency as there was a column ‘gender’ which was functionally dependent (indirectly in relationship with ‘name’) on ‘user_id’ and ‘name’. So we created a new table by splitting the original table into ‘user ’ and ‘user_gender’.**

**USER TABLE BEFORE 3NF:**

```
CREATE TABLE `user` (
  `user_id` int(11) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  `age` varchar(45) DEFAULT NULL,
  `gender` varchar(45) DEFAULT NULL,
  `email` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`user_id`)
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;


**USER TABLE AFTER 3NF:**

```
CREATE TABLE `users` (
   `user_id` INT(11) NOT NULL,
   `name` VARCHAR(45) DEFAULT NULL,
   `age` VARCHAR(45) DEFAULT NULL,
   `email` VARCHAR(45) DEFAULT NULL,
   PRIMARY KEY (`user_id`)
)  ENGINE=INNODB DEFAULT CHARSET=UTF8;

INSERT INTO users(user_id, name, age, email) SELECT user_id, name, age, email FROM
user;

CREATE TABLE `user_gender` (
   `user_id` INT(11) NOT NULL,
   `gender` VARCHAR(45) DEFAULT NULL,
   CONSTRAINT `user_id_gender_fk` FOREIGN KEY (`user_id`)
      REFERENCES `users` (`user_id`)
)  ENGINE=INNODB DEFAULT CHARSET=UTF8;

INSERT INTO user_gender(user_id, gender) SELECT user_id, gender FROM user;

ALTER TABLE bookings
DROP FOREIGN KEY user_id;

DROP TABLE user;

ALTER TABLE users
RENAME user;
```

```
ALTER TABLE bookings
ADD CONSTRAINT user_id
FOREIGN KEY (user_id) REFERENCES user(user_id);
```

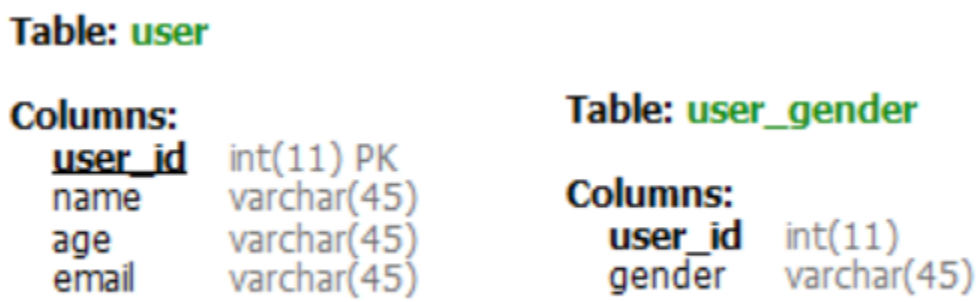
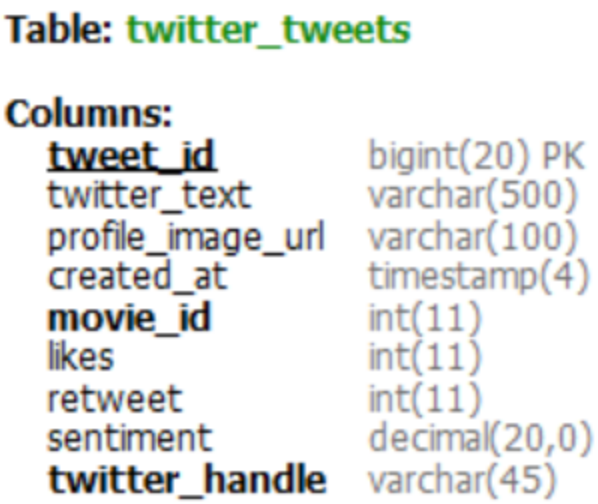



**TABLE TWITTER TWEETS:**

**Table 'Twitter Tweets' had Transitive Dependency. So we created two new tables by splitting the original table into 'twitter_tweets', 'twitter_tweet_handle' and 'twitter_tweet_movie'.**

**TWITTER TWEETS TABLE BEFORE 3NF:**

```
CREATE TABLE `twitter_tweets` (
  `tweet_id` bigint(20) NOT NULL,
  `twitter_text` varchar(500) DEFAULT NULL,
  `profile_image_url` varchar(100) CHARACTER SET utf8 DEFAULT NULL,
  `created_at` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE
CURRENT_TIMESTAMP(4),
  `movie_id` int(11) DEFAULT NULL,
  `likes` int(11) DEFAULT NULL,
  `retweet` int(11) DEFAULT NULL,
  `sentiment` decimal(20,0) DEFAULT NULL,
  `twitter_handle` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`tweet_id`),
  KEY `movie_id_tweets_idx` (`movie_id`),
  KEY `twitter_handle_fk_idx` (`twitter_handle`),
  CONSTRAINT `movie_id_tweets` FOREIGN KEY (`movie_id`) REFERENCES `movie`
(`movie_id`),
  CONSTRAINT `twitter_handle_fk` FOREIGN KEY (`twitter_handle`) REFERENCES
`twitter_user` (`twitter_handle`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```




**TWITTER TWEETS TABLE AFTER 3NF:**

```
CREATE TABLE `twitter_tweet` (
  `tweet_id` bigint(20) NOT NULL,
  `twitter_text` varchar(500) DEFAULT NULL,
  `profile_image_url` varchar(100) CHARACTER SET utf8 DEFAULT NULL,
  `created_at` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE
CURRENT_TIMESTAMP(4),
  `likes` int(11) DEFAULT NULL,
  `retweet` int(11) DEFAULT NULL,
  `sentiment` decimal(20,0) DEFAULT NULL,
  PRIMARY KEY (`tweet_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO twitter_tweet(tweet_id, twitter_text, profile_image_url, created_at, likes, retweet,
sentiment) SELECT tweet_id, twitter_text, profile_image_url, created_at, likes, retweet,
sentiment FROM twitter_tweets;

CREATE TABLE `twitter_tweet_movie` (
    `tweet_id` bigint(20) NOT NULL,
    `movie_id` int(11) DEFAULT NULL,
          KEY `movie_id_tweet_idx` (`movie_id`),
          CONSTRAINT `movie_id_tweet` FOREIGN KEY (`movie_id`) REFERENCES `movie`
(`movie_id`)
)  ENGINE=INNODB DEFAULT CHARSET=utf8mb4;

INSERT INTO twitter_tweet_movie(tweet_id, movie_id) SELECT tweet_id, movie_id FROM
twitter_tweets;
```

```
CREATE TABLE `twitter_tweet_handle` (
	`tweet_id` bigint(20) NOT NULL,
	`twitter_handle` varchar(45) DEFAULT NULL,
	KEY `twitter_handles__fk_idx` (`twitter_handle`),
	CONSTRAINT `twitter__handles_fk` FOREIGN KEY (`twitter_handle`) REFERENCES
`twitter_user` (`twitter_handle`)
)  ENGINE=INNODB DEFAULT CHARSET=utf8mb4;

INSERT INTO twitter_tweet_handle(tweet_id, twitter_handle) SELECT tweet_id, twitter_handle
FROM twitter_tweets;

SET FOREIGN_KEY_CHECKS = 0;

DROP TABLE twitter_tweets;

ALTER TABLE twitter_tweet
RENAME twitter_tweets;

SET FOREIGN_KEY_CHECKS = 1;

ALTER TABLE twitter_tweet_movie
ADD CONSTRAINT tweet_id_fk
FOREIGN KEY (tweet_id) REFERENCES twitter_tweets(tweet_id);

ALTER TABLE twitter_tweet_handle
ADD CONSTRAINT tweet_id_fks
FOREIGN KEY (tweet_id) REFERENCES twitter_tweets(tweet_id);
```


**Table:** twitter_tweets

**Columns:**

| | |
|---|---|
| **tweet_id** | bigint(20) PK |
| twitter_text | varchar(500) |
| profile_image_url | varchar(100) |
| created_at | timestamp(4) |
| likes | int(11) |
| retweet | int(11) |
| sentiment | decimal(20,0) |

**ALL OTHER TABLES:**

As you can see, there are no duplicate rows in the following tables and no repeating groups so all the following tables are in 1NF. And as there are no Partial Dependencies in the following tables the tables are in 2NF. And as there are no Transitive Dependencies in the following tables all the following tables are in compliance with the 3rd Normal Form.

**EXAMPLES:**

1) **BOOKINGS TABLE:**


| booking_id | created_at | user_id | screen_shows_id | tickets_booked |
|---|---|---|---|---|
| 1 | 2022-01-02 09:48:51.0000 | 771 | 52 | 3 |
| 2 | 2021-11-07 19:40:13.0000 | 810 | 273 | 1 |
| 3 | 2021-06-05 08:12:41.0000 | 596 | 198 | 2 |
| 4 | 2020-01-21 22:55:05.0000 | 987 | 265 | 8 |
| 5 | 2022-06-27 15:35:42.0000 | 933 | 27 | 7 |

2) **PFA OTHER TABLE SCREEN SHOTS IN GITHUB REPO FOLDER ASSIGNMENT 4 -> TABLE SCREENSHOTS.**