HiveQL Case Study- Assignment

Dataset Overview

The dataset simulates Instagram's operations and includes the following tables:

- **Dimension Tables**:

```
-- dim_user table

CREATE TABLE dim_user (

    user_id INT,

    username STRING,

    full_name STRING,

    email STRING,

    gender STRING,

    date_of_birth DATE,

    signup_date DATE,

    country STRING,

    language STRING

);


-- Create Table: dim_user

CREATE TABLE dim_user (

    user_id INT,

    username STRING,

    full_name STRING,
    email STRING,
    gender STRING,
    date_of_birth DATE,
    signup_date DATE,
    country STRING,
    language STRING
);

-- Insert Sample Data into dim_user
INSERT INTO dim_user VALUES
(1, 'john_doe', 'John Doe', 'john.doe@example.com', 'Male', '1995-06-15', '2020-01-10', 'USA', 'English'),
(2, 'jane_smith', 'Jane Smith', 'jane.smith@example.com', 'Female', '1998-04-20', '2021-03-15', 'Canada', 'English'),
(3, 'mike_jones', 'Mike Jones', 'mike.jones@example.com', 'Male', '1992-11-30', '2019-06-25', 'UK', 'English'),
(4, 'emma_wilson', 'Emma Wilson', 'emma.wilson@example.com', 'Female', '1997-09-18', '2022-01-05', 'Australia', 'English'),
(5, 'alex_brown', 'Alex Brown', 'alex.brown@example.com', 'Non-binary', '1993-02-14', '2020-07-22', 'USA', 'English'),
(6, 'chris_evans', 'Chris Evans', 'chris.evans@example.com', 'Male', '1990-12-04', '2021-09-10', 'USA', 'English'),
(7, 'olivia_adams', 'Olivia Adams', 'olivia.adams@example.com', 'Female', '1999-01-23', '2022-05-18', 'Canada', 'French'),
(8, 'li_wang', 'Li Wang', 'li.wang@example.com', 'Male', '1994-07-08', '2018-11-12', 'China', 'Mandarin'),
(9, 'sophia_lee', 'Sophia Lee', 'sophia.lee@example.com', 'Female', '1996-03-19', '2020-02-27', 'South Korea', 'Korean'),
(10, 'david_kim', 'David Kim', 'david.kim@example.com', 'Male', '1991-05-03', '2019-10-30',
'South Korea', 'Korean');



-- dim_post table
CREATE TABLE dim_post (
    post_id INT,
    user_id INT,
    caption STRING,
    post_date DATE,
    post_time STRING,
    location_id INT,
    device_id INT
);
INSERT INTO dim_post (post_id, user_id, caption, post_date, post_time, location_id,
device_id) VALUES
(1, 101, 'Sunset at the beach', '2024-12-25', '18:30', 201, 301),
(2, 102, 'Delicious homemade pizza', '2024-12-24', '19:00', 202, 302),
(3, 103, 'Hiking trip in the mountains', '2024-12-23', '08:00', 203, 303),
(4, 104, 'Birthday celebration with friends', '2024-12-22', '20:00', 204, 304),
(5, 105, 'Family picnic at the park', '2024-12-21', '12:00', 205, 305),
(6, 106, 'Concert night with my favorite band', '2024-12-20', '21:00', 206, 306),
(7, 107, 'Exploring the city at night', '2024-12-19', '22:00', 207, 307),
(8, 108, 'Weekend getaway to the countryside', '2024-12-18', '10:00', 208, 308),
(9, 109, 'Art exhibition visit', '2024-12-17', '15:00', 209, 309),
(10, 110, 'Learning to play guitar!', '2024-12-16', '17:00', 210, 310);


-- dim_hashtag table
CREATE TABLE dim_hashtag (
    hashtag_id INT,
    hashtag_text STRING
);
INSERT INTO dim_hashtag (hashtag_id, hashtag_text) VALUES
(1, '#sunset'),
(2, '#foodie'),
(3, '#hiking'),
(4, '#birthday'),
(5, '#familytime'),
(6, '#concert'),
(7, '#cityexploration'),
(8, '#getaway'),
(9, '#art'),
(10, '#guitar');
```

```
-- dim_location table
CREATE TABLE dim_location (
    location_id INT,
    location_name STRING,
    city STRING,
    country STRING,
    latitude DOUBLE,
    longitude DOUBLE
);
INSERT INTO dim_location (location_id, location_name, city, country, latitude, longitude) VALUES
(201, 'Sunny Beach', 'Miami', 'USA', 25.7617, -80.1918),
(202, 'Pizzeria Italia', 'New York', 'USA', 40.7128, -74.0060),
(203, 'Mountain Peak Trail', 'Denver', 'USA', 39.7392, -104.9903),
(204, 'Central Park', 'New York', 'USA', 40.7851, -73.9683),
(205, 'City Park', 'Los Angeles', 'USA', 34.0522, -118.2437),
(206, 'Stadium Concert Hall', 'Chicago', 'USA', 41.8781, -87.6298),
(207, 'Downtown District', 'San Francisco', 'USA', 37.7749, -122.4194),
(208, 'Countryside Retreat', 'Austin', 'USA', 30.2672, -97.7431),
(209, 'Art Gallery District', 'Seattle', 'USA', 47.6062,-122.3321),
(210,'Music School','Boston','USA','42.3601','-71.0589');
```

```
-- dim_device table

CREATE TABLE dim_device (

    device_id INT,

    device_type STRING,

    os STRING,

    os_version STRING

);

INSERT INTO dim_device (device_id, device_type, os, os_version) VALUES

(301,'Smartphone','Android','11'),

(302,'Tablet','iOS','14'),

(303,'Laptop','Windows','10'),

(304,'Smartwatch','Wear OS','2.0'),

(305,'Desktop PC','Linux','Ubuntu 20.04'),

(306,'Smartphone','iOS','15'),

(307,'E-reader','Kindle OS','5.13'),

(308,'Gaming Console','PlayStation OS','8.0'),

(309,'Smart TV','Android TV','9'),

(310,'Virtual Reality Headset','VR OS','1.0');
```

  - `dim_user`
  - `dim_post`
  - `dim_hashtag`
  - `dim_location`
  - `dim_device`
- **Fact Tables**:

```
-- fact_posts table

CREATE TABLE fact_posts (
    post_id INT,
    user_id INT,
    caption STRING,
    post_date DATE,
    post_time STRING,
    location_id INT,
    device_id INT,
    media_type STRING,
    likes_count INT,
    comments_count INT,
    shares_count INT
);


INSERT INTO fact_posts (post_id, user_id, caption, post_date, post_time, location_id, device_id, media_type, likes_count, comments_count, shares_count) VALUES
(1, 101, 'Sunset at the beach', '2024-12-25', '18:30', 201, 301, 'Image', 150, 10, 5),
(2, 102, 'Delicious homemade pizza', '2024-12-24', '19:00', 202, 302, 'Image', 200, 20, 15),
(3, 103, 'Hiking trip in the mountains', '2024-12-23', '08:00', 203, 303, 'Image', 120, 5, 3),
(4, 104, 'Birthday celebration with friends', '2024-12-22', '20:00', 204, 304, 'Video', 300, 50, 25),
(5, 105, 'Family picnic at the park', '2024-12-21', '12:00', 205, 305, 'Image', 80, 12, 7),
(6, 106, 'Concert night with my favorite band', '2024-12-20', '21:00', 206, 306, 'Video', 250, 30, 20),
(7, 107, 'Exploring the city at night', '2024-12-19', '22:00', 207, 307, 'Image', 90, 8, 4),
(8, 108, 'Weekend getaway to the countryside', '2024-12-18', '10:00', 208, 308, 'Image', 110, 15, 10),
(9, 109, 'Art exhibition visit', '2024-12-17', '15:00', 209, 309, 'Image', 75, 6, 2),
(10, 110, 'Learning to play guitar!', '2024-12-16', '17:00', 210, 310,'Video' ,130 ,14 ,8);
-- fact_likes table
CREATE TABLE fact_likes (
    like_id INT,
    post_id INT,
    user_id INT,
    like_date DATE,
    like_time STRING
);
INSERT INTO fact_likes (like_id, post_id, user_id, like_date, like_time) VALUES
(1, 1 ,101,'2024-12-25','18:35'),
(2 ,2 ,102,'2024-12-24','19:05'),
(3 ,3 ,103,'2024-12-23','08:10'),
(4 ,4 ,104,'2024-12-22','20:05'),
(5 ,5 ,105,'2024-12-21','12:05'),
(6 ,6 ,106,'2024-12-20','21:05'),
(7 ,7 ,107,'2024-12-19','22:05'),
(8 ,8 ,108,'2024-12-18','10:05'),
(9 ,9 ,109,'2024-12-17','15:05'),
(10 ,10 ,110,'2024-12-16','17:05');
```

```
-- fact_comments table
CREATE TABLE fact_comments (
    comment_id INT,
    post_id INT,
    user_id INT,
    comment_text STRING,
    comment_date DATE,
    comment_time STRING
);

INSERT INTO fact_comments (comment_id, post_id,user_id ,comment_text ,comment_date ,comment_time) VALUES
(1 ,1 ,102 ,'Beautiful view!' ,'2024-12-25' ,'18:40'),
(2 ,2 ,103 ,'Looks delicious!' ,'2024-12-24' ,'19:10'),
(3 ,3 ,104 ,'Wish I was there!' ,'2024-12-23' ,'08:15'),
(4 ,4 ,105 ,'Happy birthday!' ,'2024-12-22' ,'20:10'),
(5 ,5 ,106 ,'Sounds fun!' ,'2024-12-21' ,'12:10'),
(6 ,6 ,107 ,'What a great concert!' ,'2024-12-20' ,'21:10'),
(7 ,7 ,108 ,'Love exploring cities!' ,'2024-12-19' ,'22:10'),
(8 ,8 ,109 ,'Countryside looks peaceful.' ,'2024-12-18' ,'10:10'),
(9 ,9 ,110 ,'Art is life!' ,'2024-12-17' ,'15:10'),
(10 ,10 ,101 ,'Keep practicing!' ,'2024-12-16' ,'17:10');
```

```
-- fact_followers table
CREATE TABLE fact_followers (
    follower_user_id INT,
    followed_user_id INT,
    follow_date DATE
);

INSERT INTO fact_followers (follower_user_id, followed_user_id, follow_date) VALUES
(2010 ,101,'2023 -11 -01'),
(2011 ,102,'2023 -11 -02'),
(2012 ,103,'2023 -11 -03'),
(2013 ,104,'2023 -11 -04'),
(2014 ,105,'2023 -11 -05'),
(2015 ,106,'2023 -11 -06'),
(2016 ,107,'2023 -11 -07'),
(2017 ,108,'2023 -11 -08'),
(2018 ,109,'2023 -11 -09'),
(2019 ,110,'2023 -11 -10');


-- fact_user_activity table
```

```
CREATE TABLE fact_user_activity (
    activity_id INT,
    user_id INT,
    activity_type STRING,
    activity_date DATE,
    activity_time STRING,
    device_id INT
);

INSERT INTO fact_user_activity (activity_id,user_id ,activity_type ,activity_date ,
activity_time ,device_id) VALUES
(1 ,101 ,'Post Created' ,'2024 -12 -25' ,'18 :30' ,
301),
(2 ,102 ,'Post Liked' ,'2024 -12 -24' ,'19 :00' ,
302),
(3 ,103 ,'Commented on Post' ,'2024 -12 -23' ,'08 :00' ,
303),
(4 ,104 ,'Followed User' ,'2024 -12 -22' ,'20 :00' ,
304),
(5 ,105 ,'Post Shared' ,'2024 -12 -21' ,'12 :00' ,
305),
(6 ,106 ,'Post Created' ,'2024 -12 -20' ,'21 :00' ,
306),
(7 ,107 ,'Commented on Post' ,'2024 -12 -19' ,'22 :00' ,
307),
(8 ,108 ,'Post Liked' ,'2024 -12 -18' ,'10 :00' ,
308),
(9 ,109 ,'Followed User' ,'2024 -12 -17' ,'15 :00',
309),
(10 ,110,'Post Created','2024 -12 -16','17 :00',
310);
```

- `fact_posts`
- `fact_likes`
- `fact_comments`
- `fact_followers`
- `fact_user_activity`

1. **`dim_user`**:
    - **Columns**: `user_id`, `username`, `full_name`, `email`, `gender`, `date_of_birth`, `signup_date`, `country`, `language`
    - **Description**: Contains user profile information.
2. **`dim_post`**:
    - **Columns**: `post_id`, `user_id`, `caption`, `post_date`, `post_time`, `location_id`, `device_id`
    - **Description**: Contains metadata about each post.
3. **`dim_hashtag`**:
    - **Columns**: `hashtag_id`, `hashtag_text`
    - **Description**: Contains unique hashtags used in posts.
4. **`dim_location`**:
    - **Columns**: `location_id`, `location_name`, `city`, `country`, `latitude`, `longitude`
    - **Description**: Contains information about locations tagged in posts.
5. **`dim_device`**:
    - **Columns**: `device_id`, `device_type`, `os`, `os_version`

    - **Description**: Contains information about devices used to access Instagram.

6. `fact_posts`:
    - **Columns**: `post_id`, `user_id`, `caption`, `post_date`, `post_time`, `location_id`, `device_id`, `media_type`, `likes_count`, `comments_count`, `shares_count`
    - **Description**: Fact table containing posts made by users.
7. `fact_likes`:
    - **Columns**: `like_id`, `post_id`, `user_id`, `like_date`, `like_time`
    - **Description**: Records of likes given by users to posts.
8. `fact_comments`:
    - **Columns**: `comment_id`, `post_id`, `user_id`, `comment_text`, `comment_date`, `comment_time`
    - **Description**: Records of comments made by users on posts.
9. `fact_followers`:
    - **Columns**: `follower_user_id`, `followed_user_id`, `follow_date`
    - **Description**: Records of follow relationships between users.
10. `fact_user_activity`:
    - **Columns**: `activity_id`, `user_id`, `activity_type`, `activity_date`, `activity_time`, `device_id`
    - **Description**: Records of user activities such as login, logout, post creation, etc.

Question 1: Calculate the Top 5 Most Active Users

Business Problem:

Identify the top 5 users who have made the most posts in the last month.

Requirements:

- Use a **window function** to rank users.
- Filter posts from the last month.
- Present the user's username, full name, total posts, and rank.

Output Example:

username	full_name	total_posts	user_rank
john_doe	John Doe	25	1
jane_doe	Jane Doe	20	2
alex_smith	Alex Smith	15	3
maria_gonz	Maria Gonz	10	4
sarah_jane	Sarah Jane	8	5

Question 2: Analyze Hashtag Popularity

Business Problem:

Determine the top 10 most used hashtags in posts over the past week.

Requirements:

- Use **temporary tables** to handle the mapping of posts to hashtags.
- Assume there is a mapping table `fact_post_hashtags` with columns `post_id`, `hashtag_id`.
- Present the hashtag text and usage count.

- **Output Example:**

hashtag_text	usage_count
#sunset	50
#coffee	40
#nature	35
#travel	30
#morning	25
#fitness	20
#photography	18
#food	15
#selfie	12
#happy	10

Question 3: Identify Influencers with High Engagement

Business Problem:

Find users who have more than 10,000 followers and an average post engagement (likes + comments) greater than 500 in the past month.

Requirements:

- Use **CTEs** to calculate follower counts and average engagement.
- Present the user's username, follower count, average engagement, and total posts.

Output Example:

username	follower_count	avg_engagement	total_posts
john_doe	12000	550	30
jane_doe	11000	530	25

Question 4: Standardize Device Information Using a UDF

Business Problem:

Ensure all device operating systems in `dim_device` are standardized to uppercase (e.g., 'ios' becomes 'IOS').

Requirements:

- Create and use a **UDF** called `to_upper_case`.
- Update the `dim_device` table with standardized OS names.
- Show a sample of updated device records.

---

device_id	device_type	os	os_version
301	Phone	IOS	14.2
302	Phone	ANDROID	11.0

Question 5: Calculate User Retention Rate

Business Problem:

Determine the retention rate of users who signed up in the last 6 months and are still active.

Requirements:

- Use **window functions** to calculate retention.
- Define active users as those who have logged in within the past month.
- Present the total number of new users and the number of active users.

Output Example:

total_new_users	total_active_users	retention_rate_percent

1000	600	60.00

Question 6: Use Bucketing to Sample User Activity

Business Problem:

Analyze a 10% sample of user activities to test a new feature without processing the entire dataset.

Requirements:

- Utilize **bucketing** on `user_id` to efficiently sample data.
- Use the `TABLESAMPLE` clause.
- Present a count of activities in the sample.

Output Example:

count
15000

Question 7: Create a View for Users with Incomplete Profiles

Business Problem:

Identify users who have not completed their profiles (missing email or date of birth) for a targeted completion campaign.

Requirements:

- Use the `CREATE VIEW` statement.
- Include user ID, username, and missing fields.
- Present a sample of the view.

Output Example:

user_id	username	missing_email	missing_dob
1	john_doe	Missing Email	NULL
2	jane_doe	NULL	Missing Date of Birth

Question 8: Determine Average Comments per Post per Category

Business Problem:

Calculate the average number of comments per post for different media types (e.g., photo, video, story).

Requirements:

- Use **window functions** or **GROUP BY**.
- Present the media type and average comments.
- Order results by average comments in descending order.

Output Example:

media_type	avg_comments
video	20.5
photo	15.8
story	10.2

Question 9: Identify Posts with High Engagement Using CTEs and Window Functions

Business Problem:

Find posts that are in the top 1% in terms of engagement (likes + comments) over the past week.

Requirements:

- Use **CTEs** and **window functions** to calculate the engagement percentile.
- Present post ID, user ID, engagement score, and percentile.
- Filter for posts in the 99th percentile or higher.

Output Example:

post_id	user_id	engagement_score	engagement_percentile

101	1	300	1.00
102	2	290	0.995

Question 10: Mask Sensitive Data in Comments Using a UDF

Business Problem:

For data privacy, create a report of comments where any email addresses mentioned are masked.

Requirements:

- Create and use a **UDF** called `mask_emails_in_text`.
- Present comment ID, user ID, and masked comment text.
- Ensure that email addresses within the comment text are replaced with '[email protected]'.

Output Example:

comment_id	user_id	masked_comment_text
201	1	"Contact me at [email protected]"

202	2	"Email me at [email protected]"