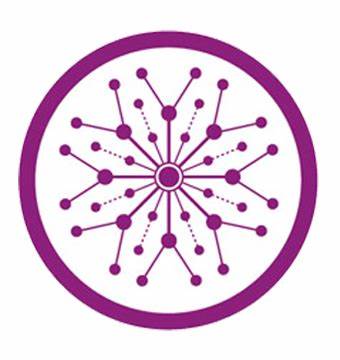
**THE SUPERIOR UNIVERSITY, LAHORE**

****

Department of Information Engineering Technology

Faculty of Engineering & Technology

Project Report:

**[Social Media Website Schema]**

**[University admission management system]**

**[Financial Database Schema]**

**Database Management System Projects:**

**Project Team:**

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Student ID** | **Program** |
| Usmana Zulfiqar | BIET-F21-013 | Bs-IET |
| Manahil Tehseen | BIET-F21-067 | Bs-IET |
| Zoha Mehmood | BIET-F21-017 | Bs-IET |
| Zunaisha Noor | BIET-F21-049 | Bs-IET |

**Social Media Schema:**

**Introduction:**

Social media schema refers to the design and organization of the database that stores social media data. A social media schema defines the structure and relationships of the various types of data that are stored in the database, including user profiles, posts, comments, likes, and shares.

The main role of a social media schema is to ensure that the data is organized and stored in a way that enables efficient querying and retrieval. A well-designed schema can also help to ensure the integrity and consistency of the data by enforcing data constraints, such as data types and relationships between entities.

Some of the key components of a social media schema might include tables for storing user data (such as user ID, name, email, and password), tables for storing post data (such as post ID, content, date, and author), and tables for storing interactions between users and posts (such as likes, comments, and shares).

The design of a social media schema will depend on the specific requirements of the social media platform and the types of data that need to be stored and queried. In general, however, a well-designed social media schema will be optimized for performance, scalability, and data integrity, and will be able to handle large volumes of data and complex queries.

**Objectives:**

The objectives of a social media schema can be summarized as follows:

1. **Data organization**: The primary objective of a social media schema is to organize the data in a way that allows for efficient querying and retrieval. This involves structuring the data into tables and defining relationships between them to facilitate data access.
2. **Data integrity:** Another important objective of a social media schema is to ensure the accuracy and consistency of the data stored in the database. This is achieved by enforcing data constraints such as data types, ranges, and relationships between entities.
3. **Performance**: A well-designed social media schema should be optimized for performance, which includes minimizing query response times, reducing storage requirements, and ensuring scalability.
4. **Security**: The schema should be designed to protect sensitive data and prevent unauthorized access to the system. This can be achieved by defining access control policies and encrypting data.
5. **Maintainability**: The schema should be designed to make maintenance and upgrades easier. This involves using standardized naming conventions, avoiding redundancy, and minimizing dependencies.
6. **Flexibility**: A social media schema should be designed to be flexible and adaptable to changing requirements. This means that it should be easy to modify the schema as the needs of the system evolve.

**Database schema with entity and attributes:**

Designing a database schema for a social media website system can be a complex task as it depends on various factors like the functionalities and features of the social media platform, the number of users, and the type of data that needs to be stored. However, below is an example of a possible database schema that can be used as a starting point for a social media website system:

1. **User Table:**

* user\_id (Primary Key)
* username
* password
* email
* full\_name
* bio
* profile\_picture
* date\_joined

1. **Friends Table:**

* friendship\_id (Primary Key)
* user\_id (Foreign Key to User Table)
* friend\_id (Foreign Key to User Table)
* status (Pending, Accepted, Rejected)

1. **Posts Table:**

* post\_id (Primary Key)
* user\_id (Foreign Key to User Table)
* text
* media (image, video, etc.)
* date\_created

1. **Comments Table:**

* comment\_id (Primary Key)
* user\_id (Foreign Key to User Table)
* post\_id (Foreign Key to Posts Table)
* text
* date\_created

1. **Likes Table:**

* like\_id (Primary Key)
* user\_id (Foreign Key to User Table)
* post\_id (Foreign Key to Posts Table)
* date\_created

1. **Groups Table:**

* group\_id (Primary Key)
* group\_name
* group\_description
* date\_created

1. **Group Members Table:**

* membership\_id (Primary Key)
* user\_id (Foreign Key to User Table)
* group\_id (Foreign Key to Groups Table)
* date\_joined

1. **Messages Table:**

* message\_id (Primary Key)
* sender\_id (Foreign Key to User Table)
* receiver\_id (Foreign Key to User Table)
* text
* date\_se

**Complete code with queries:**

CREATE TABLE User\_table(

user\_id VARCHAR2(10) PRIMARY KEY,

username VARCHAR(50) NOT NULL unique,

password VARCHAR2(50) NOT NULL,

email VARCHAR2(100) NOT NULL unique,

full\_name VARCHAR2(100) NOT NULL,

bio VARCHAR(500),

date\_joined DATE DEFAULT SYSDATE

);

SELECT User\_Table.username, User\_Table.full\_name, Posts.text AS post\_text,

Comments.text AS comment\_text, Comments.date\_created

FROM User\_Table

NATURAL JOIN Posts

NATURAL JOIN Comments;

select \*from user\_table;

1. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u1','zunaisha', 'abc123', 'zunaisha@gmail.com', 'zunaisha noor', 'I am an IT Engineer');
2. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u2','usmana', 'usmana123', 'usmana@gmail.com', 'usmana zulfiqar', 'Software Developer');
3. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u3','manahil', 'qwerty', 'manahil@gmail.com', 'manahil tehseen', 'Graphic Designer');
4. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u4','zoha', 'spiderman', 'zoha@gmail.com', 'zoha mahmood', 'Journalist');
5. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u5','clarakent', 'superman', 'clarakent@gmail.com', 'Clark Kent', 'Reporter');
6. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u6','brucewayne', 'batman', 'brucewayne@gmail.com', 'Bruce Wayne', 'CEO');
7. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u7','tonystark', 'ironman', 'tonystark@gmail.com', 'Tony Stark', 'Inventor');
8. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u8','dianaprince', 'wonderwoman', 'dianaprince@gmail.com', 'Diana Prince', 'Amazon Warrior');
9. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u9','clarkkent', 'superman', 'clarkkent@gmail.com', 'Clark Kent', 'Reporter');
10. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u10','harveydent', 'twoface', 'harveydent@gmail.com', 'Harvey Dent', 'District Attorney');
11. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u11','oliverqueen', 'arrow', 'oliverqueen@gmail.com', 'Oliver Queen', 'Vigilante');
12. INSERT INTO user\_table (user\_id,username, password, email, full\_name, bio) VALUES ('u12','jamesbond', '007', 'jamesbond@gmail.com', 'James Bond', 'Secret Agent');

DROP table user\_table;

UPDATE USER\_TABLE set username='ali' where user\_id='u5';

alter table user\_table add gender;

select \*from user\_table where full\_name like 'z%';

select \* from posts where posts.user\_id= user\_table.user\_id;

SELECT Comments.comment\_id, Posts.post\_id, Comments.text, Comments.date\_created

FROM Comments

inner JOIN Posts on comments.user\_id = Posts.user\_id;

SELECT \*

FROM Comments

NATURAL JOIN User\_table

WHERE User\_table.user\_id = 'u4';

CREATE TABLE Posts (

post\_id VARCHAR2(10) PRIMARY KEY,

user\_id VARCHAR2(10) NOT NULL,

text VARCHAR2(1000) NOT NULL,

date\_created DATE DEFAULT SYSDATE,

FOREIGN KEY (user\_id) REFERENCES User\_table(user\_id)

);

select \*from posts;

----Retrieve all posts and their associated user information:---

SELECT Posts.post\_id, Posts.text,

Posts.date\_created, User\_table.user\_id, User\_table.username, User\_table.email, User\_table.full\_name,

User\_table.bio, User\_table.date\_joined

FROM posts

inner JOIN User\_table

ON Posts.user\_id = User\_table.user\_id;

select \*from user\_table where full\_name like 'z%';

select \* from posts where posts.user\_id= user\_table.user\_id like 'z%';

---Retrieve all comments and their associated post and user information:

SELECT Comments.comment\_id, Comments.text, Comments.date\_created,

Posts.post\_id, Posts.text,

Posts.date\_created AS post\_date\_created, User\_table.user\_id,

User\_table.username, User\_table.email, User\_table.full\_name,

User\_table.bio,User\_table.date\_joined

FROM Comments

INNER JOIN Posts

ON Comments.post\_id = Posts.post\_id

INNER JOIN User\_table

ON Comments.user\_id = User\_table.user\_id;

sELECT Posts.post\_id, Posts.text,

Posts.date\_created AS post\_date\_created, User\_table.user\_id,

User\_table.username, User\_table.email, User\_table.full\_name,

User\_table.bio,User\_table.date\_joined

FROM Posts

INNER JOIN User\_table

ON user\_table.user\_id = Posts.user\_id where user\_table.user\_id='u4' ;

select \*from posts where user\_id='u10';

1. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p1','u1', 'I just climbed Mount Everest!');
2. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p2','u2', 'I love reading books!');
3. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p3','u3', 'I am learning how to code in Python.');
4. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p4','u4', 'I had a great day at the beach!');
5. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p5','u5', 'I just adopted a new dog!');
6. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p6','u6', 'I am excited to start my new job next week!');
7. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p7','u7', 'I finished reading "To Kill a Mockingbird" and loved it!');
8. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p8','u8', 'I just ran my first marathon!');
9. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p9','u9' 'I am trying to learn how to play the guitar.');
10. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p10','u10', 'I just watched an amazing movie called "The Shawshank Redemption".');
11. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p11','u11', 'I just finished writing my first novel!');
12. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p12','u12', 'I am on vacation in Hawaii and it is beautiful!');
13. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p13','u13', 'I just completed a 30-day yoga challenge!');
14. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p14','u14', 'I just got engaged to my partner!');
15. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p15','u15', 'I am excited to start my graduate program in the fall.');
16. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p16','u16', 'I am working on a new art project.');
17. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p17','u17', 'I just started learning how to cook Indian cuisine.');
18. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p18','u18', 'I am taking a break from social media for a month.');
19. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p19','u19', 'I just finished redecorating my bedroom.');
20. INSERT INTO posts(post\_id,user\_id, text) VALUES ('p20','u20', 'I am excited to see my favorite band in concert next month.');

CREATE TABLE Comments (

comment\_id VARCHAR2(10) PRIMARY KEY,

user\_id VARCHAR2(10) NOT NULL,

post\_id VARCHAR2(10) NOT NULL,

text VARCHAR2(1000) NOT NULL,

date\_created DATE DEFAULT SYSDATE,

FOREIGN KEY (user\_id) REFERENCES User\_table(user\_id),

FOREIGN KEY (post\_id) REFERENCES Posts(post\_id)

);

select \*from comments;

--to get a list of all users and their posts,

--including posts that have not been commented on yet

SELECT u.username, p.text, c.text

FROM User\_table u

inner JOIN Posts p ON u.user\_id = p.user\_id

LEFT JOIN Comments c ON p.post\_id = c.post\_id

ORDER BY u.username, p.date\_created, c.date\_created;

1. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c1','u1', 'p1', 'Wow, that sounds amazing!');
2. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c2','u2', 'p2', 'I completely agree with you!');
3. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c3','u3', 'p3', 'Thanks for sharing your thoughts.');
4. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c4','u4', 'p4', 'This is really helpful, thank you!');
5. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c5','u5', 'p5', 'Great post, keep up the good work!');
6. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c6','u6', 'p6', 'I found this very informative.');
7. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c7','u7', 'p7', 'I can relate to this so much!');
8. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c8','u8', 'p8', 'I had no idea about this, thanks for sharing.');
9. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c9','u9', 'p9', 'Interesting perspective.');
10. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c10','u10', 'p10', 'I appreciate the effort you put into this post.');
11. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c11','u11', 'p11', 'This is so inspiring, thank you for sharing.');
12. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c12','u12', 'p12', 'I had a similar experience, thanks for sharing yours.');
13. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c13','u13', 'p13', 'I love this idea!');
14. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c14','u14', 'p14', 'I never thought of it that way before, thanks for the insight.');
15. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c15','u15', 'p15', 'I appreciate your honesty in sharing your story.');
16. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c16','u16', 'p16', 'This is a great reminder, thank you for sharing.');
17. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c17','u17', 'p17', 'I can t wait to try this out!');
18. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c18','u18', 'p18', 'I am definitely bookmarking this post for future reference.');
19. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c19','u19', 'p19', 'I enjoyed reading this, thank you for sharing your experience.');
20. INSERT INTO comments(comment\_id,user\_id, post\_id, text) VALUES ('c20','u20', 'p20', 'This is such an important topic, thank you for bringing attention to it.');

CREATE TABLE Friends (

friendship\_id VARCHAR2(10) PRIMARY KEY,

user\_id VARCHAR2(10) NOT NULL,

friend\_id VARCHAR2(10) NOT NULL,

status VARCHAR2(20) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES User\_table (user\_id),

FOREIGN KEY (friend\_id) REFERENCES User\_table (user\_id)

);

select \*from friends;

1. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_1','u1','f1','Pending');
2. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_2','u2','f2','Pending');
3. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_3','u3','f3','Pending');
4. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_4','u4','f4','Pending');
5. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_5','u5','f5','Pending');
6. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_6','u6','f6','Pending');
7. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_7','u7','f7','Pending');
8. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_8','u8','f8','Pending');
9. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_9','u9','f9','Pending');
10. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_10','u10','f10','Pending');
11. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_11','u11','f11','Pending');
12. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_12','u12','f12','Pending');
13. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_13','u13','f13','Pending');
14. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_14','u14','f14','Pending');
15. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_15','u15','f15','Pending');
16. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_16','u16','f16','Pending');
17. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_17','u17','f17','Pending');
18. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_18','u18','f18','Pending');
19. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_19','u19','f19','Pending');
20. INSERT INTO friends (friendship\_id,user\_id, friend\_id,status) VALUES ('fs\_20','u20','f20','Pending');

select\* from friends where friendship\_id like user\_id='u4';

alter table friends add gender;

CREATE TABLE Messages (

message\_id VARCHAR2(10) PRIMARY KEY,

user\_id VARCHAR2(10) NOT NULL,

sender\_id VARCHAR2(10) NOT NULL,

receiver\_id VARCHAR2(10) NOT NULL,

date\_sent DATE DEFAULT SYSDATE,

text VARCHAR2(1000) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES User\_table(user\_id),

FOREIGN KEY (user\_id) REFERENCES User\_table(user\_id)

);

select \*from messages;

1. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m1','u1','s1','r1', 'Hey, how are you doing?','01-april-2023');
2. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m2','u2','s2','r2', 'Just wanted to say hi!','02-april-2023');
3. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m3','u3','s3','r3', 'Can we catch up soon?','03-april-2023');
4. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m4','u4','s4','r4', 'How was your weekend?','04-april-2023');
5. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m5','u5','s5','r5', 'Hey, did you get my email?','05-april-2023');
6. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m6','u6','s6','r6', 'I need your help with something.','06-april-2023');
7. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m7','u7','s7','r7', 'What do you think about the new project?','07-april-2023');
8. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m8','u8','s8','r8', 'Let’s meet for coffee tomorrow!','08-april-2023');
9. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m9','u9','s9','r9', 'Do you want to grab lunch today?','09-april-2023');
10. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m10','u10','s10','r10', 'I have some news to share with you.','10-april-2023');
11. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m11','u11','s11','r11', 'Can we reschedule our meeting?','11-april-2023');
12. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m12','u12','s12','r12', 'Are you free for a call later today?','12-april-2023');
13. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m13','u13','s13','r13', 'I need your opinion on something.','13-april-2023');
14. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m14','u14','s14','r14', 'Just checking in on you.','14-april-2023');
15. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m15','u15','s15','r15', 'Hey, how are you doing?','01-april-2023');
16. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m16','u16','s16','r16', 'What are your plans for the weekend?','02-april-2023');
17. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m17','u17','s17','r17', 'I cant wait for our vacation next month!','03-april-2023');
18. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m18','u18','s18','r18', 'Do you want to grab lunch later?','04-april-2023');
19. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m19','u19','s19','r19', 'Thanks for the birthday wishes!','05-april-2023');
20. INSERT INTO messages(message\_id,user\_id,sender\_id, receiver\_id, text, date\_sent) VALUES ('m20','u20','s20','r20', 'Im so proud of you for finishing your project!','06-april-2023');

select \* from posts;

**--To retrieve all the posts with their corresponding users' information**

SELECT User\_Table.username, User\_Table.full\_name, Posts.text,Posts.date\_created

FROM User\_Table

INNER JOIN Posts

ON User\_Table.user\_id = Posts.user\_id;

**--To retrieve all the users with their corresponding posts**

SELECT User\_Table.username, User\_Table.full\_name, Posts.text,

Posts.date\_created

FROM User\_Table

LEFT OUTER JOIN Posts

ON User\_Table.user\_id = Posts.user\_id;

--user 1 ki post

select user\_table.username,

**--To retrieve all the comments with their corresponding posts' and users' information:**

SELECT User\_Table.username, User\_Table.full\_name, Posts.text AS post\_text,

Comments.text AS comment\_text, Comments.date\_created

FROM User\_Table

NATURAL JOIN Posts

NATURAL JOIN Comments;

**--to get a list of all users and their posts,**

**--including posts that have not been commented on yet**

SELECT u.username, p.text, c.text

FROM User\_table u

LEFT JOIN Posts p ON u.user\_id = p.user\_id

LEFT JOIN Comments c ON p.post\_id = c.post\_id

ORDER BY u.username, p.date\_created, c.date\_created;

CREATE TABLE groups (

group\_id VARCHAR2(10) PRIMARY KEY,

group\_name VARCHAR2(100) NOT NULL,

group\_description VARCHAR2(500) NOT NULL,

date\_created DATE DEFAULT SYSDATE

);

select \*from groups;

drop table groups;

1. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g1','Hiking Enthusiasts', 'A group for people who love hiking and exploring nature.');
2. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g2','Book Club', 'A group for avid readers to discuss and share their favorite books.');
3. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g3','Art Appreciation Society', 'A group for people who enjoy art and want to learn more about it.');
4. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g4','Cooking Club', 'A group for foodies to share recipes and cooking tips.');
5. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g5','Fitness Fanatics', 'A group for people who are passionate about fitness and living a healthy lifestyle.');
6. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g6','Tech Talk', 'A group for technology enthusiasts to discuss and share their knowledge.');
7. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g7','Language Exchange', 'A group for people who want to learn and practice different languages.');
8. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g8','Travel Junkies', 'A group for people who love to travel and explore different cultures.');
9. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g9','Film Buffs', 'A group for movie lovers to discuss and analyze their favorite films.');
10. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g10','Gardening Club', 'A group for people who enjoy gardening and growing plants.');
11. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g11','Photography Enthusiasts', 'A group for people who love photography and want to learn more about it.');
12. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g12','Writers Workshop', 'A group for aspiring writers to share their work and get feedback.');
13. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g13','Entrepreneurial Network', 'A group for entrepreneurs to connect and share ideas.');
14. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g14','Chess Club', 'A group for people who enjoy playing chess.');
15. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g15','Outdoor Adventure', 'A group for thrill-seekers who love outdoor activities.');
16. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g16','Fashionistas', 'A group for people who are interested in fashion and style.');
17. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g17','History Buffs', 'A group for people who love history and want to learn more about it.');
18. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g18','Pet Lovers', 'A group for people who love pets and want to share their experiences.');
19. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g19','Music Lovers', 'A group for people who enjoy listening to music and discussing it.');
20. INSERT INTO groups(group\_id,group\_name, group\_description) VALUES ('g20','Yoga and Meditation', 'A group for people who practice yoga and meditation.');

CREATE TABLE Group\_Members (

membership\_id VARCHAR2(10) PRIMARY KEY,

user\_id VARCHAR2(10) NOT NULL,

group\_id VARCHAR2(10) NOT NULL,

date\_joined DATE DEFAULT SYSDATE,

FOREIGN KEY (user\_id) REFERENCES User\_table(user\_id),

FOREIGN KEY (group\_id) REFERENCES Groups(group\_id)

);

select \*from Group\_Members;

1. INSERT INTO Group\_Members(membership\_id ,user\_id, group\_id) VALUES ('ms\_1','u1', 'g1');
2. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_2', 'u2', 'g2');
3. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_3', 'u3', 'g3');
4. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_4', 'u4', 'g4');
5. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_5', 'u5', 'g5');
6. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_6', 'u6', 'g6');
7. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_7', 'u7', 'g7');
8. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_8', 'u8', 'g8');
9. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_9', 'u9', 'g9');
10. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_10', 'u10', 'g10');
11. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_11', 'u11', 'g11');
12. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_12', 'u12', 'g12');
13. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_13', 'u13', 'g13');
14. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_14', 'u14', 'g14');
15. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_15', 'u15', 'g15');
16. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_16', 'u16', 'g16');
17. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_17', 'u17', 'g17');
18. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_18', 'u18', 'g18');
19. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_19', 'u19', 'g19');
20. INSERT INTO Group\_Members(membership\_id, user\_id, group\_id) VALUES ('ms\_20', 'u20', 'g20');

DROP TABLE gROUP\_MEMBER;

select \* from Group\_Members where membership\_id like '%97';search

**University Admission Management system:**

A university admission management system is a computer-based system used by universities to manage the admission process of prospective students. The system typically includes several tables, each with specific information about the admission process.

**Tables:**

1. **Student**: This table contains information about the students who are applying to the university, such as their personal details, contact information, and academic history.
2. **Programs**: This table lists all the academic programs offered by the university, such as undergraduate and graduate programs, and includes information about the program requirements and the duration of the program.
3. **Courses**: This table contains information about the courses offered by the university, including course codes, names, descriptions, and credit hours.
4. Admission Requirements: This table lists the admission requirements for each program, such as minimum GPA, standardized test scores, and prerequisite courses.
5. **Test Scores:** This table contains information about the standardized test scores submitted by the student, such as SAT, ACT, GRE, GMAT, and TOEFL scores.
6. **Application:** This table contains information about the application submitted by the student, including the date of submission, the program applied for, and the admission decision.
7. **Transcript:** This table contains information about the student's academic transcript, including their grades, GPA, and academic history.
8. **Decision:** This table contains information about the admission decision made by the university, such as the status of the application (accepted, rejected, waitlisted), and any conditions or requirements for admission.

**Complete code with queries:**

drop table student;

drop table programs;

drop table courses;

drop table admission\_requirement;

DROP TABLE TesT\_Score;

drop table Application;

drop table transcript;

drop table decision;

CREATE TABLE Student (

applicant\_id INT NOT NULL PRIMARY KEY,

name varchar2(10),

email varchar2(20),

date\_of\_birth DATE DEFAULT SYSDATE

);

select \*from student;

1. INSERT INTO student (applicant\_id, name, email) VALUES (1, 'Ali', 'ali@123.com');
2. INSERT INTO student (applicant\_id, name, email) VALUES (2, 'Fatima', 'fatima@455.com');
3. INSERT INTO student (applicant\_id, name, email) VALUES (3, 'Ahmed', 'ahmed@123.com');
4. INSERT INTO student (applicant\_id, name, email) VALUES (4, 'Aisha', 'aisha@123.com');
5. INSERT INTO student (applicant\_id, name, email) VALUES (5, 'Abdullah', 'abdullah@123.com');
6. INSERT INTO student (applicant\_id, name, email) VALUES (6, 'Amina', 'amina@123.com');
7. INSERT INTO student (applicant\_id, name, email) VALUES (7, 'Omar', 'omar@123.com');
8. INSERT INTO student (applicant\_id, name, email) VALUES (8, 'Hafsa', 'hafsa@123.com');
9. INSERT INTO student (applicant\_id, name, email) VALUES (9, 'Yusuf', 'yusuf@123.com');
10. INSERT INTO student (applicant\_id, name, email) VALUES (10, 'Khadija', 'khadija@123.com');
11. INSERT INTO student (applicant\_id, name, email) VALUES (11, 'Ibrahim', 'ibrahim@123.com');
12. INSERT INTO student (applicant\_id, name, email) VALUES (12, 'Zainab', 'zainab@123.com');
13. INSERT INTO student (applicant\_id, name, email) VALUES (13, 'Mustafa', 'mustafa@123.com');
14. INSERT INTO student (applicant\_id, name, email) VALUES (14, 'Safia', 'safia@123.com');
15. INSERT INTO student (applicant\_id, name, email) VALUES (15, 'Zaid', 'zaid@123.com');
16. INSERT INTO student (applicant\_id, name, email) VALUES (16, 'Zahra', 'zahra@123.com');
17. INSERT INTO student (applicant\_id, name, email) VALUES (17, 'Hassan', 'hassan@123.com');
18. INSERT INTO student (applicant\_id, name, email) VALUES (18, 'Maryam', 'maryam@123.com');
19. INSERT INTO student (applicant\_id, name, email) VALUES (19, 'Aliyah', 'aliyah@123.com');
20. INSERT INTO student (applicant\_id, name, email) VALUES (20, 'Abdul', 'abdul@123.com');

program\_id VARCHAR(5) PRIMARY KEY,

program\_name VARCHAR(50)

);

select \*from programs;

1. INSERT INTO programs (program\_id, program\_name) VALUES ('p1', 'Bachelor of Science in Computer Science');
2. INSERT INTO programs VALUES ('p2', 'Bachelor of Science in Information Technology');
3. INSERT INTO programs VALUES ('p3', 'Bachelor of Science in Business Administration');
4. INSERT INTO programs VALUES ('p4', 'Bachelor of Science in Marketing');
5. INSERT INTO programs VALUES ('p5', 'Bachelor of Science in Accounting');
6. INSERT INTO programs VALUES ('p6', 'Bachelor of Science in Nursing');
7. INSERT INTO programs VALUES ('p7', 'Bachelor of Science in Education');
8. INSERT INTO programs VALUES ('p8', 'Master of Science in Computer Science');
9. INSERT INTO programs VALUES ('p9', 'Master of Science in Information Technology');
10. INSERT INTO programs VALUES ('p10', 'Master of Business Administration');
11. INSERT INTO programs VALUES ('p11', 'Master of Science in Marketing');
12. INSERT INTO programs VALUES ('p12', 'Master of Science in Accounting');
13. INSERT INTO programs VALUES ('p13', 'Master of Science in Nursing');
14. INSERT INTO programs VALUES ('p14', 'Master of Education');
15. INSERT INTO programs VALUES ('p15', 'Doctor of Philosophy in Computer Science');
16. INSERT INTO programs VALUES ('p16', 'Doctor of Philosophy in Information Technology');
17. INSERT INTO programs VALUES ('p17', 'Doctor of Business Administration');
18. INSERT INTO programs VALUES ('p18', 'Doctor of Philosophy in Marketing');
19. INSERT INTO programs VALUES ('p19', 'Doctor of Philosophy in Accounting');
20. INSERT INTO programs VALUES ('p20', 'Doctor of Philosophy in Nursing');

CREATE TABLE courses (

course\_id varchar(5) PRIMARY KEY,

course\_name VARCHAR(50),

description VARCHAR(255)

);

select \*from courses;

1. INSERT INTO courses (course\_id, course\_name, description) VALUES ('p1', 'Introduction to Computer Science', 'An introduction to the principles of computer science and programming.');
2. INSERT INTO courses VALUES ('p2', 'Introduction to Computer Science', 'An introduction to the fundamental concepts of computer science.');
3. INSERT INTO courses VALUES ('p3', 'Operating Systems', 'A study of operating system concepts, design, and implementation.');
4. INSERT INTO courses VALUES ('p4', 'Computer Networks', 'A study of computer networks, protocols, and network architecture.');
5. INSERT INTO courses VALUES ('p5', 'Database Systems', 'A study of database design, implementation, and management.');
6. INSERT INTO courses VALUES ('p6', 'Artificial Intelligence', 'A study of artificial intelligence concepts, algorithms, and applications.');
7. INSERT INTO courses VALUES ('p7', 'Machine Learning', 'A study of machine learning algorithms and their applications.');
8. INSERT INTO courses VALUES ('p8', 'Computer Graphics', 'A study of computer graphics principles and techniques.');
9. INSERT INTO courses VALUES ('p9', 'Software Engineering', 'A study of software engineering concepts and methodologies.');
10. INSERT INTO courses VALUES ('p10', 'Programming Languages', 'A study of programming language design, implementation, and semantics.');
11. INSERT INTO courses VALUES ('p11', 'Computer Security', 'A study of computer security threats, vulnerabilities, and countermeasures.');
12. INSERT INTO courses VALUES ('p12', 'Web Development', 'A study of web development technologies, frameworks, and design principles.');
13. INSERT INTO courses VALUES ('p13', 'Mobile Application Development', 'A study of mobile application development frameworks and design principles.');
14. INSERT INTO courses VALUES ('p14', 'Human-Computer Interaction', 'A study of human-computer interaction principles and design methodologies.');
15. INSERT INTO courses VALUES ('p15', 'Data Mining', 'A study of data mining algorithms and their applications.');
16. INSERT INTO courses VALUES ('p16', 'Big Data', 'A study of big data analytics technologies and methodologies.');
17. INSERT INTO courses VALUES ('p17', 'Computer Vision', 'A study of computer vision algorithms and applications.');
18. INSERT INTO courses VALUES ('p18', 'Natural Language Processing', 'A study of natural language processing techniques and applications.');
19. INSERT INTO courses VALUES ('p19', 'Cloud Computing', 'A study of cloud computing concepts, architectures, and technologies.');
20. INSERT INTO courses VALUES ('p20', 'Parallel Computing', 'A study of parallel computing architectures and algorithms.');

CREATE TABLE admission\_requirement (

admission\_id VARCHAR(5) PRIMARY KEY,

program\_id VARCHAR(5),

admission\_type VARCHAR(50),

FOREIGN KEY (program\_id) REFERENCES programs(program\_id)

);

select \*from admission\_requirement;

1. INSERT INTO admission\_requirement (admission\_id, program\_id, admission\_type) VALUES ('a1', 'p1', 'diploma or equivalent');
2. INSERT INTO admission\_requirement VALUES ('a2', 'p1', 'Alevel');
3. INSERT INTO admission\_requirement VALUES ('a3', 'p1', 'ICs');
4. INSERT INTO admission\_requirement VALUES ('a4', 'p2', 'diploma or equivalent');
5. INSERT INTO admission\_requirement VALUES ('a5', 'p2', 'ICs');
6. INSERT INTO admission\_requirement VALUES ('a6', 'p2', 'ICs');
7. INSERT INTO admission\_requirement VALUES ('a7', 'p3', 'diploma or equivalent');
8. INSERT INTO admission\_requirement VALUES ('a8', 'p3', 'pre-engineering');
9. INSERT INTO admission\_requirement VALUES ('a9', 'p3', 'Intermediate');
10. INSERT INTO admission\_requirement VALUES ('a10', 'p4', 'diploma or equivalent');
11. INSERT INTO admission\_requirement VALUES ('a11', 'p4', 'Ics');
12. INSERT INTO admission\_requirement VALUES ('a12', 'p4', 'pre-medical');
13. INSERT INTO admission\_requirement VALUES ('a13', 'p5', ' diploma or equivalent');
14. INSERT INTO admission\_requirement VALUES ('a14', 'p5', 'pre-medical');
15. INSERT INTO admission\_requirement VALUES ('a15', 'p5', 'pre-medical');
16. INSERT INTO admission\_requirement VALUES ('a16', 'p6', 'diploma or equivalent');
17. INSERT INTO admission\_requirement VALUES ('a17', 'p6', 'pre-medical');
18. INSERT INTO admission\_requirement VALUES ('a18', 'p6', 'ICom');
19. INSERT INTO admission\_requirement VALUES ('a19', 'p7', 'diploma or equivalent');
20. INSERT INTO admission\_requirement VALUES ('a20', 'p7', 'pre-medical');

CREATE TABLE Test\_Score (

Score\_ID VARCHAR(6) PRIMARY KEY,

Test\_Type VARCHAR(50) NOT NULL,

Score INT NOT NULL,

applicant\_id INT NOT NULL,

FOREIGN KEY (applicant\_id) REFERENCES Student(applicant\_id)

);

select \* from Test\_Score;

1. INSERT INTO Test\_Score (Score\_ID, Test\_Type, Score,applicant\_id) VALUES ('S1', 'MCQS', 1056,1);
2. INSERT INTO Test\_Score VALUES ('S2', 'Q/A', 956,2);
3. INSERT INTO Test\_Score VALUES ('S3', 'MULTIPLE CHOICE', 76,3);
4. INSERT INTO Test\_Score VALUES ('S4', 'SHORT ANSWER', 94,4);
5. INSERT INTO Test\_Score VALUES ('S5', 'TRUE/FALSE', 88,5);
6. INSERT INTO Test\_Score VALUES ('S6', 'MCQS', 980,6);
7. INSERT INTO Test\_Score VALUES ('S7', 'ESSAY', 76,7);
8. INSERT INTO Test\_Score VALUES ('S8', 'MULTIPLE CHOICE', 82,8);
9. INSERT INTO Test\_Score VALUES ('S9', 'SHORT ANSWER', 88,9);
10. INSERT INTO Test\_Score VALUES ('S10', 'TRUE/FALSE', 94,10);
11. INSERT INTO Test\_Score VALUES ('S11', 'MCQS', 940,11);
12. INSERT INTO Test\_Score VALUES ('S12', 'ESSAY', 92,12);
13. INSERT INTO Test\_Score VALUES ('S13', 'MULTIPLE CHOICE', 80,13);
14. INSERT INTO Test\_Score VALUES ('S14', 'SHORT ANSWER', 90,14);
15. INSERT INTO Test\_Score VALUES ('S15', 'TRUE/FALSE', 86,15);
16. INSERT INTO Test\_Score VALUES ('S16', 'MCQS', 880,16);
17. INSERT INTO Test\_Score VALUES ('S17', 'ESSAY', 95,17);
18. INSERT INTO Test\_Score VALUES ('S18', 'MULTIPLE CHOICE', 72,18);
19. INSERT INTO Test\_Score VALUES ('S19', 'SHORT ANSWER', 98,19);
20. INSERT INTO Test\_Score VALUES ('S20', 'TRUE/FALSE', 90,20);

create table Application (

Application\_ID VARCHAR(50) PRIMARY KEY,

program\_id VARCHAR(50) NOT NULL ,

Status VARCHAR(50) NOT NULL,

Applicant\_ID INT NOT NULL,

Application\_Date DATE DEFAULT SYSDATE,

FOREIGN KEY (applicant\_id) REFERENCES Student(applicant\_id),

FOREIGN KEY (Program\_ID) REFERENCES programs(program\_id)

);

select \*from Application;

1. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap1', 'p1', 'approved', 1);
2. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap2', 'p2', 'rejected', 2);
3. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap3', 'p3', 'pending', 3);
4. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap4', 'p4', 'approved', 4);
5. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap5', 'p5', 'rejected', 5);
6. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap6', 'p6', 'pending', 6);
7. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap7', 'p7', 'approved', 7);
8. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap8', 'p8', 'rejected', 8);
9. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap9', 'p9', 'pending', 9);
10. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap10', 'p10', 'approved', 10);
11. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap11', 'p11', 'rejected', 11);
12. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap12', 'p12', 'pending', 12);
13. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap13', 'p13', 'approved', 13);
14. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap14', 'p14', 'rejected', 14);
15. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap15', 'p15', 'pending', 15);
16. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap16', 'p16', 'approved', 16);
17. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap17', 'p17', 'rejected', 17);
18. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap18', 'p18', 'pending', 18);
19. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap19', 'p19', 'approved', 19);
20. INSERT INTO Application (Application\_ID, Program\_ID, Status, Applicant\_ID) VALUES ('ap20', 'p20', 'rejected', 20);

CREATE TABLE transcript (

transcript\_id VARCHAR(50) PRIMARY KEY,

degree\_type VARCHAR(50),

completion\_year INT,

percentage DECIMAL(5,2),

FOREIGN KEY (applicant\_id) REFERENCES student(applicant\_id)

);

select \*from transcript;

1. INSERT INTO transcript (transcript\_id, applicant\_id, degree\_type, completion\_year, percentage) VALUES ('t1', 'a1', 'diploma', 2020, 87.5);
2. INSERT INTO transcript VALUES ('t2', 'a2', 'intermediate', 2019, 91.3);
3. INSERT INTO transcript VALUES('t3', 'a3', 'diploma', 2021, 79.8);
4. INSERT INTO transcript VALUE ('t4', 'a4', 'intermediate', 2018, 82.4);
5. INSERT INTO transcript VALUES ('t5', 'a5', 'diploma', 2022, 90.1);
6. INSERT INTO transcript VALUES ('t6', 'a6', 'intermediate', 2017, 75.2);
7. INSERT INTO transcript VALUES ('t7', 'a7', 'diploma', 2023, 86.7);
8. INSERT INTO transcript VALUES ('t8', 'a8', 'intermediate', 2016, 93.2);
9. INSERT INTO transcript VALUES ('t9', 'a9', 'diploma', 2020, 80.5);
10. INSERT INTO transcript VALUES ('t10', 'a10', 'intermediate', 2019, 88.9);
11. INSERT INTO transcript VALUES ('t11', 'a11', 'diploma', 2021, 92.1);
12. INSERT INTO transcript VALUES ('t12', 'a12', 'intermediate', 2018, 70.3);
13. INSERT INTO transcript VALUES ('t13', 'a13', 'diploma', 2022, 83.6);
14. INSERT INTO transcript VALUES ('t14', 'a14', 'intermediate', 2017, 91.8);
15. INSERT INTO transcript VALUES ('t15', 'a15', 'diploma', 2023, 76.5);
16. INSERT INTO transcript VALUES ('t16', 'a16', 'intermediate', 2016, 84.9);
17. INSERT INTO transcript VALUES ('t17', 'a17', 'diploma', 2020, 89.2);
18. INSERT INTO transcript VALUES ('t18', 'a18', 'intermediate', 2019, 78.6);
19. INSERT INTO transcript VALUES ('t19', 'a19', 'diploma', 2021, 85.1);
20. INSERT INTO transcript VALUES ('t20', 'a20', 'intermediate', 2018, 92.3);

CREATE TABLE decision (

decision\_id VARCHAR(50) PRIMARY KEY,

decision\_status VARCHAR(20) NOT NULL,

applicant\_id INT NOT NULL,

FOREIGN KEY (applicant\_id) REFERENCES student(applicant\_id)

);

drop table decision;

select \*from decision;

1. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES ('d1', 'approved', 1);
2. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d2', 'rejected', 2);
3. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d3', 'pending', 3);
4. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d4', 'approved', 4);
5. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d5', 'rejected', 5);
6. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d6', 'pending', 6);
7. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d7', 'approved', 7);
8. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d8', 'rejected', 8);
9. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d9', 'pending', 9);
10. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d10', 'approved', 10);
11. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d11', 'rejected', 11);
12. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d12', 'pending', 12);
13. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d13', 'approved', 13);
14. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d14', 'rejected', 14);
15. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d15', 'pending', 15);
16. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d16', 'approved', 16);
17. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d17', 'rejected', 17);
18. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d18', 'pending', 18);
19. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d19', 'approved', 19);
20. INSERT INTO decision (decision\_id, decision\_status, applicant\_id) VALUES('d20', 'rejected', 20);

**select \*from student where name like 'l%';**

**select \*from student where name like '%r';**

**select \*from student where name like '%a%';**

**--- Inner join to get all applications along with their corresponding admission decisions:**

SELECT \* FROM Application

INNER JOIN decision

ON Application.applicant\_id = decision.applicant\_id;

**----Outer join to get all applications, even if there is no corresponding admission decision:**

```SELECT \*

FROM Application

LEFT OUTER JOIN Decision

ON Application.applicant\_id = Decision.applicant\_id;

**Financial Database Schema:**

A financial database schema is a database design that is specifically optimized for storing financial data. The objective of a financial database schema is to efficiently and accurately store, manage, and retrieve financial data for various purposes, such as reporting, analysis, and decision-making.

**Tables:**

The tables, keys, and attributes of a financial database schema will vary depending on the specific requirements of the application, but here are some common examples:

1. **Accounts Table:** This table stores information about financial accounts, such as account number, account type, balance, and owner information.
2. **Transactions Table:** This table stores information about financial transactions, such as transaction date, transaction type, amount, and account information.
3. **Customers Table**: This table stores information about customers, such as name, contact information, and account information.
4. **Financial Institutions Table**: This table stores information about financial institutions, such as name, contact information, and routing information.
5. **Currency Table**: This table stores information about currencies, such as currency code, currency name, and exchange rate.
6. **Security Table**: This table stores information about financial securities, such as security type, symbol, price, and performance information.

**Purpose:**

The purpose and objective of a financial database schema are to efficiently and accurately store, manage, and retrieve financial data for various purposes, such as reporting, analysis, and decision-making.

Financial data is a critical asset for many organizations, including banks, financial institutions, accounting firms, and businesses that manage financial transactions. A financial database schema helps these organizations store financial data in a structured way that can be easily accessed and analyzed.

**Objectives:**

The primary objectives of a financial database schema include:

* **Accurately storing financial data:** A financial database schema is designed to accurately store financial data with a high degree of precision and reliability.
* **Facilitating efficient data retrieval**: The schema should allow for efficient retrieval of data based on specific criteria, such as account number, transaction date, or customer information.
* **Supporting complex queries and reporting:** A financial database schema should be optimized for complex queries and reporting, which may involve aggregating and analyzing large volumes of data.
* **Ensuring data integrity and security**: The schema should ensure that financial data is protected from unauthorized access, while also ensuring that the data is accurate and consistent.
* **Facilitating integration with other systems:** The schema should be designed to facilitate integration with other systems, such as accounting software, financial reporting tools, or customer relationship management system.

**COMPLETE CODE WITH QUERIES:**

drop table customers;

drop table categories;

drop table accounts;

drop table Transactions;

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(50),

customer\_address VARCHAR(100),

customer\_phone VARCHAR(20),

customer\_email VARCHAR(50)

);

1. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (1, 'John Smith', '123 Main St', '555-1234', 'john.smith@email.com');
2. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (2, 'Jane Doe', '456 Elm St', '555-5678', 'jane.doe@email.com');
3. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (3, 'Bob Johnson', '789 Oak St', '555-9012', 'bob.johnson@email.com');
4. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (4, 'Sara Lee', '101 Cherry Ave', '555-3456', 'sara.lee@email.com');
5. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (5, 'Mike Smith', '555 Pine St', '555-6789', 'mike.smith@email.com');
6. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (6, 'Tina Turner', '888 Maple Ave', '555-1234', 'tina.turner@email.com');
7. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (7, 'David Brown', '222 Cedar St', '555-5678', 'david.brown@email.com');
8. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (8, 'Lisa Johnson', '333 Oak St', '555-9012', 'lisa.johnson@email.com');
9. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (9, 'Chris Davis', '444 Main St', '555-3456', 'chris.davis@email.com');
10. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (10, 'Karen White', '777 Pine St', '555-6789', 'karen.white@email.com');
11. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (11, 'Mark Wilson', '999 Maple Ave', '555-1234', 'mark.wilson@email.com');
12. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (12, 'Amy Adams', '111 Cedar St', '555-5678', 'amy.adams@email.com');
13. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (13, 'Tim Taylor', '222 Oak St', '555-9012', 'tim.taylor@email.com');
14. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (14, 'Jenny Lee', '333 Main St', '555-3456', 'jenny.lee@email.com');
15. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (15, 'Tom Jones', '444 Pine St', '555-6789', 'tom.jones@email.com');
16. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (16, 'Rachel Green', '555 Maple Ave', '555-1234', 'rachel.green@email.com');
17. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (17, 'Chandler Bing', '777 Elm St', '555-5678', 'chandler.bing@email.com');
18. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (18, 'Monica Geller', '888 Oak St', '555-9012', 'monica.geller@email.com');
19. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (19, 'Ross Geller', '999 Cedar St', '555-3456', 'ross.geller@email.com');
20. INSERT INTO Customers (customer\_id, customer\_name, customer\_address, customer\_phone, customer\_email) VALUES (20, 'Phoebe Buffay', '111 Main St', '555-6789', 'phoebe.buffay@email.com');

select\*from Customers;

SELECT \* FROM Customers WHERE customer\_id = <insert\_customer\_id>;

SELECT customer\_name, customer\_email FROM Customers

INNER JOIN Accounts ON Customers.customer\_id = Accounts.customer\_id

WHERE Accounts.account\_id = <insert\_account\_id>;

SELECT \* FROM Customers WHERE customer\_id IN (

SELECT customer\_id FROM Accounts WHERE account\_balance > <insert\_balance\_amount>);

SELECT customer\_name, SUM(account\_balance) as total\_balance FROM Customers

INNER JOIN Accounts ON Customers.customer\_id = Accounts.customer\_id

GROUP BY customer\_name;

CREATE TABLE Categories (

category\_id INT PRIMARY KEY,

category\_name VARCHAR(50)

);

1. INSERT INTO Categories (category\_id, category\_name) VALUES (1, 'Electronics');
2. INSERT INTO Categories (category\_id, category\_name) VALUES (2, 'Clothing');
3. INSERT INTO Categories (category\_id, category\_name) VALUES (3, 'Home and Garden');
4. INSERT INTO Categories (category\_id, category\_name) VALUES (4, 'Beauty and Personal Care');
5. INSERT INTO Categories (category\_id, category\_name) VALUES (5, 'Sports and Outdoors');
6. INSERT INTO Categories (category\_id, category\_name) VALUES (6, 'Toys and Games');
7. INSERT INTO Categories (category\_id, category\_name) VALUES (7, 'Automotive');
8. INSERT INTO Categories (category\_id, category\_name) VALUES (8, 'Books and Audible');
9. INSERT INTO Categories (category\_id, category\_name) VALUES (9, 'Music');
10. INSERT INTO Categories (category\_id, category\_name) VALUES (10, 'Movies and TV Shows');
11. INSERT INTO Categories (category\_id, category\_name) VALUES (11, 'Grocery and Gourmet Food');
12. INSERT INTO Categories (category\_id, category\_name) VALUES (12, 'Pet Supplies');
13. INSERT INTO Categories (category\_id, category\_name) VALUES (13, 'Health and Wellness');
14. INSERT INTO Categories (category\_id, category\_name) VALUES (14, 'Industrial and Scientific');
15. INSERT INTO Categories (category\_id, category\_name) VALUES (15, 'Office Products');
16. INSERT INTO Categories (category\_id, category\_name) VALUES (16, 'Software and Computer Games');
17. INSERT INTO Categories (category\_id, category\_name) VALUES (17, 'Baby Products');
18. INSERT INTO Categories (category\_id, category\_name) VALUES (18, 'Arts, Crafts, and Sewing');
19. INSERT INTO Categories (category\_id, category\_name) VALUES (19, 'Cell Phones and Accessories');
20. INSERT INTO Categories (category\_id, category\_name) VALUES (20, 'Jewelry and Watches');

select\*from Categories;

SELECT \* FROM Categories WHERE category\_id = <insert\_category\_id>;

CREATE TABLE Accounts (

account\_id INT PRIMARY KEY,

account\_type VARCHAR(20),

account\_balance DECIMAL(10,2),

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

1. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (1, 'Checking', 5000.00, 1);
2. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (2, 'Savings', 10000.00, 1);
3. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (3, 'Checking', 2500.00, 2);
4. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (4, 'Savings', 7500.00, 2);
5. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (5, 'Checking', 10000.00, 3);
6. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (6, 'Savings', 15000.00, 3);
7. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (7, 'Checking', 7500.00, 4);
8. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (8, 'Savings', 12500.00, 4);
9. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (9, 'Checking', 12500.00, 5);
10. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (10, 'Savings', 17500.00, 5);
11. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (11, 'Checking', 3000.00, 6);
12. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (12, 'Savings', 6000.00, 6);
13. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (13, 'Checking', 8000.00, 7);
14. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (14, 'Savings', 12000.00, 7);
15. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (15, 'Checking', 10000.00, 8);
16. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (16, 'Savings', 15000.00, 8);
17. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (17, 'Checking', 7500.00, 9);
18. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (18, 'Savings', 12500.00, 9);
19. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (19, 'Checking', 20000.00, 10);
20. INSERT INTO Accounts (account\_id, account\_type, account\_balance, customer\_id) VALUES (20, 'Savings', 30000.00, 10);

select\*from Accounts;

SELECT \* FROM Accounts WHERE account\_id = <insert\_account\_id>;

SELECT account\_id, COUNT(\*) FROM Transactions GROUP BY account\_id;

SELECT account\_type, SUM(account\_balance) as total\_balance FROM Accounts

GROUP BY account\_type;

SELECT account\_id, SUM(transaction\_amount) as total\_amount, AVG(transaction\_amount) as avg\_amount FROM Transactions

GROUP BY account\_id;

CREATE TABLE Transactions (

transaction\_id INT PRIMARY KEY,

transaction\_type VARCHAR(20),

transaction\_amount DECIMAL(10,2),

transaction\_date DATE,

account\_id INT,

FOREIGN KEY (account\_id) REFERENCES Accounts(account\_id)

);

INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (1, 'Deposit', 500.00, '2023-05-01', 1001);

INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (2, 'Withdrawal', 50.00, '2023-05-02', 1001);

INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (3, 'Transfer', 200.00, '2023-05-02', 1001);

INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (4, 'Deposit', 1000.00, '2023-05-03', 1002);

INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (5, 'Withdrawal', 250.00, '2023-05-04', 1002);

1. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (6, 'Transfer', 150.00, '2023-05-05', 1002);
2. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (7, 'Deposit', 750.00, '2023-05-01', 1003);
3. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (8, 'Withdrawal', 100.00, '2023-05-02', 1003);
4. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (9, 'Transfer', 50.00, '2023-05-02', 1003);
5. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (10, 'Deposit', 2000.00, '2023-05-03', 1004);
6. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (11, 'Withdrawal', 500.00, '2023-05-04', 1004);
7. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (12, 'Transfer', 300.00, '2023-05-05', 1004);
8. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (13, 'Deposit', 1000.00, '2023-05-01', 1005);
9. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (14, 'Withdrawal', 200.00, '2023-05-02', 1005);
10. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (15, 'Transfer', 100.00, '2023-05-02', 1005);
11. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (16, 'Deposit', 500.00, '2023-05-03', 1006);
12. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (17, 'Withdrawal', 100.00, '2023-05-04', 1006);
13. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (18, 'Transfer', 50.00, '2023-05-05', 1006);
14. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (19, 'Deposit', 1500.00, '2023-05-01', 1007);
15. INSERT INTO Transactions (transaction\_id, transaction\_type, transaction\_amount, transaction\_date, account\_id)VALUES (20, 'Withdrawal', 300.00, '2023-05-02', 1007);

select\*from transactions;

SELECT \* FROM Transactions WHERE account\_id = <insert\_account\_id>;

SELECT \* FROM Transactions WHERE transaction\_date BETWEEN <start\_date> AND <end\_date>;

SELECT SUM(account\_balance) FROM Accounts;

SELECT customer\_id, COUNT(\*) FROM Accounts GROUP BY customer\_id;

SELECT transaction\_type, AVG(transaction\_amount) FROM Transactions GROUP BY transaction\_type;

SELECT \* FROM Transactions WHERE account\_id IN (SELECT account\_id FROM Accounts WHERE customer\_id IN (

SELECT customer\_id FROM Customers WHERE customer\_email = '<insert\_email\_address>')

);

SELECT \* FROM Transactions WHERE transaction\_type = '<insert\_transaction\_type>' AND transaction\_date BETWEEN <start\_date> AND <end\_date>;